

CFM 2409 US  
09/982,916



日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年11月10日

出 願 番 号

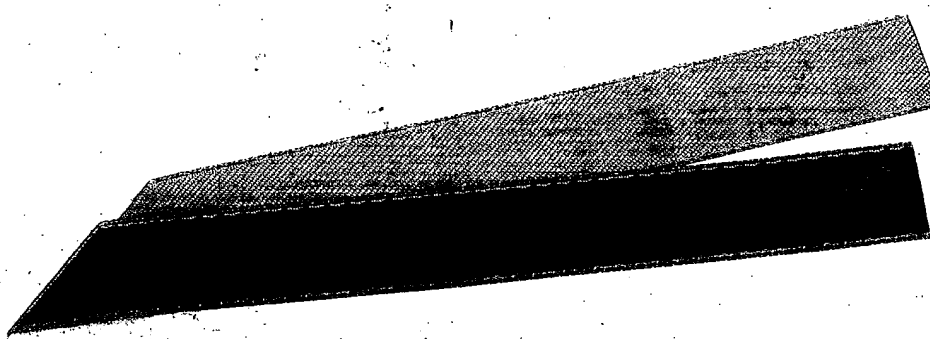
Application Number:

特願2000-344311

出 願 人

Applicant(s):

キヤノン株式会社



RECEIVED

MAR 14 2002

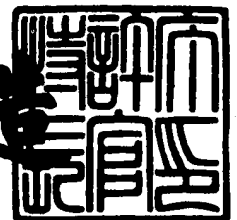
Technology Center 2100

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年11月16日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3100526

【書類名】 特許願

【整理番号】 4272057

【提出日】 平成12年11月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明の名称】 画像データ用フィルタ処理装置及びその制御方法

【請求項の数】 7

【発明者】

【住所又は居所】 東京都大田区下丸子3丁目30番2号 キヤノン株式会社  
社内

【氏名】 中山 忠義

【特許出願人】

【識別番号】 000001007

【氏名又は名称】 キヤノン株式会社

【代理人】

【識別番号】 100076428

【弁理士】

【氏名又は名称】 大塚 康徳

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100101306

【弁理士】

【氏名又は名称】 丸山 幸雄

【電話番号】 03-5276-3241

【選任した代理人】

【識別番号】 100115071

【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0001010

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像データ用フィルタ処理装置及びその制御方法

【特許請求の範囲】

【請求項 1】 フィルタ処理しようとする画素データに対応する 3 つのデータの供給を受け、出力データを演算する演算ユニットを複数備える最上位演算層と、

1 つ上の層で演算された出力データの 2 つ、及び、2 つ上の層で得られたデータの 1 つの 3 入力を受け、出力データを演算する前記演算ユニットを複数個備える中間演算層と、

前記最上位演算層と、前記中間演算層を複数積層する構造を有するフィルタ処理装置であって、

前記複数の演算ユニットのそれぞれは、

与えられた 3 つのデータを用いて出力データを演算する第 1 の演算モードを有する第 1 の演算ユニットと、

与えられた 3 つのデータ中の 2 つのデータに基づき、3 つ分のデータによる出力データを演算する第 2 の演算モードと、前記第 1 の演算モードとを切り換え可能な第 2 の演算ユニットのいずれかで構成され、

フィルタ処理しようとするデータが画像の境界近傍のタイミングで入力された場合、前記第 2 の演算ユニットの演算モードを第 2 の演算モードに切り換えることを特徴とする画像データ用フィルタ処理装置。

【請求項 2】 更に、前記最上位演算層に供給するデータの配列の順序を逆順に切り換える手段を備えることを特徴とする請求項第 1 項に記載の画像データ用フィルタ処理装置。

【請求項 3】 前記切り換え手段は、境界後端を処理するよりも前のタイミングで切り換えることを特徴とする請求項第 2 項に記載の画像データ用フィルタ処理装置。

【請求項 4】 前記第 1、第 2 の演算ユニットには、入力したデータを下位層に供給するために記憶保持するためのバッファメモリを有することを特徴とする請求項第 1 項乃至第 3 項のいずれか 1 項に記載の画像データ用フィルタ処理装

置。

【請求項 5】 ウェーブレット変換又は逆ウェーブレット変換することを特徴とする請求項第 1 項乃至第 4 項のいずれか 1 項に記載の画像データ用フィルタ処理装置。

【請求項 6】 更に、前記第 2 の演算モードを 2 種類有する第 3 の演算ユニットを備えることを特徴とする請求項第 1 項に記載の画像データ用フィルタ処理装置。

【請求項 7】 フィルタ処理しようとする画素データに対応する 3 つのデータの供給を受け、出力データを演算する演算ユニットを複数備える最上位演算層と、

1 つ上の層で演算された出力データの 2 つ、及び、2 つ上の層で得られたデータの 1 つの 3 入力を受け、出力データを演算する前記演算ユニットを複数個備える中間演算層と、

前記最上位演算層と、前記中間演算層を複数積層する構造を有するフィルタ処理装置の制御方法であって、

前記複数の演算ユニットのそれぞれは、

与えられた 3 つのデータを用いて出力データを演算する第 1 の演算モードを有する第 1 の演算ユニットと、

与えられた 3 つのデータ中の 2 つのデータに基づき、3 つ分のデータによる出力データを演算する第 2 の演算モードと、前記第 1 の演算モードとを切り換え可能な第 2 の演算ユニットのいずれかで構成され、

フィルタ処理しようとするデータが画像の境界近傍のタイミングで入力された場合、前記第 2 の演算ユニットの演算モードを第 2 の演算モードに切り換えるステップを備えることを特徴とする画像データ用フィルタ処理装置の制御方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は画像データをウェーブレット変換したり、ウェーブレット変換係数を

画像データに逆変換する等の処理に使用される画像データ用フィルタ処理装置及びその制御方法に関するものである。

#### 【 0 0 0 2 】

##### 【従来の技術】

画像、特に多値画像は非常に多くの情報を含んでおり、その画像を蓄積・伝送する際にはデータ量が膨大になるという問題がある。このため画像の蓄積・伝送に際しては、画像の持つ冗長性を除く、或いは画質の劣化が視覚的に認識し難い程度にまで画像データの変化を許容することによってデータ量を削減する高能率符号化が用いられる。

#### 【 0 0 0 3 】

例えば、静止画像の国際標準符号化方式として I S O と I T U - T により勧告された J P E G では、画像データをブロック（8画素×8画素）ごとに離散コサイン変換（D C T）して、D C T 係数に変換した後に、各係数を各々量子化し、さらにエントロピー符号化することにより画像データを圧縮している。ブロックごとに D C T、量子化を行なっているため、復号画像の各ブロックの境界で、所謂ブロック歪みが見える場合がある。

#### 【 0 0 0 4 】

一方、新しい静止画像の国際標準符号化方式として J P E G 2 0 0 0 が検討されているが、J P E G 2 0 0 0 では、量子化の前に行う前の変換処理として、ウェーブレット変換が提案されている。ウェーブレット変換は、現行 J P E G のようにブロック単位で処理を行うのではなく、入力データを連続的に処理するので、復号画像の劣化が視覚的に見えにくいといった特徴がある。

#### 【 0 0 0 5 】

J P E G 2 0 0 0 で使われているウェーブレット変換は、Lifting Scheme という方法で処理をすると、少ない演算量で効率良く変換処理を行なうことができる。

#### 【 0 0 0 6 】

図 1 に順方向の Lifting Scheme、図 2 に逆方向の Lifting Scheme のシグナルフローを表わす図を示す。図中の  $\alpha$ 、 $\beta$ 、 $\gamma$ 、 $\delta$  は Lifting 係数と呼ばれるもので

ある。以下、図 1 の動作について説明する。

【0 0 0 7】

入力画素を  $X_0, X_1, X_2, X_3, X_4, X_5, \dots$  のように順に表わす。該入力画素は、分類ユニット 2 0 1 にて、偶数画素系列と奇数画素系列とに分類され、該ユニットの上側には添字が偶数の画素  $X_0, X_2, X_4, \dots$  が、下側には添字が奇数の画素  $X_1, X_3, X_5, \dots$  が出力される。

【0 0 0 8】

初段のLifting 処理では、偶数画素系列に対しLifting係数： $\alpha$ を乗算し、連続する 2 個の乗算結果を、該 2 画素の中央に位置する奇数画素系列中の画素に加算する。

【0 0 0 9】

これを一般化した式で表現すると、以下のようになる。

【0 0 1 0】

$$D_{2n+1} = X_{2n+1} + \alpha \cdot X_{2n} + \alpha \cdot X_{2n+2} \quad (1)$$

2 段目のLifting 処理では、新たに得られた奇数画素系列  $D_1, D_3, D_5, \dots$  に対しLifting係数： $\beta$ を乗算し、連続する 2 個の乗算結果を、該 2 画素の中央に位置する偶数画素系列中の画素に加算する。

【0 0 1 1】

これを一般化した式で表現すると、以下のようになる。

【0 0 1 2】

$$E_{2n+2} = X_{2n+2} + \beta \cdot D_{2n+1} + \beta \cdot D_{2n+3} \quad (2)$$

3 段目のLifting 処理では、Lifting係数： $\gamma$ を用いて、初段と同じようにし、4 段目のLifting処理では、Lifting係数： $\delta$ を用いて、2 段目と同じように処理する。3 段、4 段目のLifting処理内容を表わす式は、下記のようになる。

【0 0 1 3】

$$H_{2n+1} = D_{2n+1} + \gamma \cdot E_{2n} + \gamma \cdot E_{2n+2} \quad (3)$$

$$L_{2n+2} = E_{2n+2} + \delta \cdot H_{2n+1} + \delta \cdot H_{2n+3} \quad (4)$$

図中、 $K$ は、ウェーブレット係数を正規化するものであるが、本発明の本質を説明するにあたって、特に関係ないことであるので、以下、説明を省略する。

## 【 0 0 1 4 】

正規化処理を無視すれば、3 段、4 段目のLifting処理によってえられる、 $H_n$ ， $L_n$ は各々高域変換係数と低域変換係数に対応する。

## 【 0 0 1 5 】

次に、図 2 に示す逆方向のLifting Schemeのシグナルフローについて簡単に説明する。まず始めに、順方向のLiftingScheme における正規化処理に対応して、逆の係数を掛けた後、4 段のLifting 処理を行なう。各段の処理内容を以下にまとめて式で表わす。

$$(1 \text{ 段目}) \quad E_{2n+2} = L_{2n+2} - \delta \cdot H_{2n+1} - \delta \cdot H_{2n+3} \quad (5)$$

$$(2 \text{ 段目}) \quad D_{2n+1} = H_{2n+1} - \gamma \cdot E_{2n} - \gamma \cdot E_{2n+2} \quad (6)$$

$$(3 \text{ 段目}) \quad X_{2n+2} = E_{2n+2} - \beta \cdot D_{2n+1} - \beta \cdot D_{2n+3} \quad (7)$$

$$(4 \text{ 段目}) \quad X_{2n+1} = D_{2n+1} - \alpha \cdot X_{2n} - \alpha \cdot X_{2n+2} \quad (8)$$

上記 (5) (6) (7) (8) 式は、各々 (4) (3) (2) (1) 式を移項して得られるものである。

## 【 0 0 1 6 】

図 1 及び図 2 のLifting Schemeを別の視点から表現したものが、図 3 及び図 4 に示すLifting格子構造である。同図において、□は入力データを、○は格子点（あるいは格子点データ演算器）を表わし、○から出ている矢印は格子点データの流れを示す。これらの図はLiftingScheme における基本処理（前記 (1) ～ (8) 式の処理）並びに該処理によって得られる新たなデータを 1 つの格子点に対応させたものである。

## 【 0 0 1 7 】

図 3 に示す順方向の Lifting 格子構造では、1 つの格子点データは前記 (1) ～ (4) 式のいずれかをを用いて計算される。また、図 4 に示す逆方向のLifting 格子構造では、1 つの格子点データは前記 (5) ～ (8) 式のいずれかにより計算される。

## 【 0 0 1 8 】

ウェーブレット変換等のフィルタ処理では、1 つの出力を得るために多くのデータを必要とするため、画像の境界部の画素に対するフィルタ出力を計算する際



には、該画像の外側の領域を参照することになるが、外側の領域にはもちろんデータが無い。

#### 【 0 0 1 9 】

そこで、該画像の境界部では通常とは異なる処理（以下では、境界処理と称す）が必要となる。最も単純な方法として、あらかじめ境界部のデータを外側に連続コピーして参照される領域にデータを書き込んでおくやり方がある。

#### 【 0 0 2 0 】

J P E G 2 0 0 0におけるウェーブレット変換処理では、境界部のデータを中心として内側のデータを外側に折り返す方法で、外側の領域にデータを配置している。

#### 【 0 0 2 1 】

図 5 に具体的な例を示す。同図において■は境界の入力データを表わし、その他は前記図 3 の表記に準ずる。

#### 【 0 0 2 2 】

左側の境界入力データである X 0 の外側（左側）には、境界の内側のデータ X 1 , X 2 , X 3 , X 4 を折り返して配置し、右側の境界入力データである X 5 の外側（右側）には、境界の内側のデータ X 4 , X 3 , X 2 を折り返して配置している。

#### 【 0 0 2 3 】

このように境界の外側にあらかじめデータを用意しておくことで、元のデータ全てに対応したウェーブレット変換係数を、例外処理をすることなく演算することができる。

#### 【 0 0 2 4 】

#### 【発明が解決しようとする課題】

以上説明した境界の外側にデータ用意する方法には、次のような問題点がある。

- ・ あらかじめ境界の外側にデータを用意するための処理をわざわざしなければならない。
- ・ 折り返して作成した境界の外側のデータを格納するための記憶領域が余分に必要となる。

【 0 0 2 5 】

本発明はかかる問題点に鑑みなされたものであり、処理しようとするデータが画像データの境界外にある場合であっても、その境界外のデータを前処理で生成する必要が無く、以って構造を簡略化させることを可能ならしめる画像データ用フィルタ処理装置及びその制御方法を提供しようとするものである。

【 0 0 2 6 】

【課題を解決するための手段】

かかる課題を解決するため、例えば本発明の画像データ用フィルタ処理装置は以下の構成を備える。すなわち、

フィルタ処理しようとする画素データに対応する3つのデータの供給を受け、出力データを演算する演算ユニットを複数備える最上位演算層と、

1つ上の層で演算された出力データの2つ、及び、2つ上の層で得られたデータの1つの3入力を受け、出力データを演算する前記演算ユニットを複数個備える中間演算層と、

前記最上位演算層と、前記中間演算層を複数積層する構造を有するフィルタ処理装置であって、

前記複数の演算ユニットのそれぞれは、

与えられた3つのデータを用いて出力データを演算する第1の演算モードを有する第1の演算ユニットと、

与えられた3つのデータ中の2つのデータに基づき、3つ分のデータによる出力データを演算する第2の演算モードと、前記第1の演算モードとを切り換え可能な第2の演算ユニットのいずれかで構成され、

フィルタ処理しようとするデータが画像の境界近傍のタイミングで入力された場合、前記第2の演算ユニットの演算モードを第2の演算モードに切り換えることを特徴とする。

【 0 0 2 7 】

【発明の実施の形態】

以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【 0 0 2 8 】

実施形態は、ウェーブレット変換処理あるいはフィルタ処理をリフティング演算で処理するときに、画像境界部の内側データを外側に折り返して利用する場合の境界処理を効率よく行なうことを可能にする。その前に、まず、用語の定義を行なう。

## 【 0 0 2 9 】

前記 (1) ～ (8) 式のいずれか 1 つのみを演算する格子点データ演算器を単機能格子点データ演算器と呼ぶ。

## 【 0 0 3 0 】

それに対して、下記に示す境界処理に対応した演算モードを併せ持つ格子点データ演算器を境界格子点データ演算器と呼ぶ。

## 【 0 0 3 1 】

$$D_0 = X_0 + 2\alpha \cdot X_1 \quad (1a)$$

$$E_0 = X_0 + 2\beta \cdot D_1 \quad (2a)$$

$$H_0 = D_0 + 2\gamma \cdot E_1 \quad (3a)$$

$$L_0 = E_0 + 2\delta \cdot H_1 \quad (4a)$$

$$D_N = X_N + 2\alpha \cdot X_{N-1} \quad (1b)$$

$$E_N = X_N + 2\beta \cdot D_{N-1} \quad (2b)$$

$$H_N = D_N + 2\gamma \cdot E_{N-1} \quad (3b)$$

$$L_N = E_N + 2\delta \cdot H_{N-1} \quad (4b)$$

$$E_0 = L_0 - 2\delta \cdot H_1 \quad (5a)$$

$$D_0 = H_0 - 2\gamma \cdot E_1 \quad (6a)$$

$$X_0 = E_0 - 2\beta \cdot D_1 \quad (7a)$$

$$X_0 = D_0 - 2\alpha \cdot X_1 \quad (8a)$$

$$E_N = L_N - 2\delta \cdot H_{N-1} \quad (5b)$$

$$D_N = H_N - 2\gamma \cdot E_{N-1} \quad (6b)$$

$$X_N = E_N - 2\beta \cdot D_{N-1} \quad (7b)$$

$$X_N = D_N - 2\alpha \cdot X_{N-1} \quad (8b)$$

上記において、式 (1a) ～ (4a) , (5a) ～ (8a) は先頭の境界データに対応する演算で、式 (1b) ～ (4b) , (5b) ～ (8b) は最後の境界データに

対応する演算である。

#### 【 0 0 3 2 】

(1) 式を演算する単機能格子点データ演算器をベースにした境界格子点データ演算器は、(1a) 式と (1b) 式のどちらか一方の演算機能を併せ持つか、あるいは両方の演算機能を併せ持つ。(2) ~ (8) 式に対応する格子点データ演算器も同様である。

#### 【 0 0 3 3 】

##### < 第 1 の実施形態 >

以上の用語の定義を踏まえ、本発明に係る第 1 の実施形態の構成を図 8 に示す。同図において、

●は境界データに対応する演算機能を併せ持つ格子点データ演算、すなわち境界格子点データ演算器であり、○は単機能格子点データ演算器である。その他は前記図 3 の表記に準ずる。

#### 【 0 0 3 4 】

E0 を出力する境界格子点データ演算器は前記 (2) 式と (2a) 式の演算機能を、L0 を出力する境界格子点データ演算器は前記 (4) 式と (4a) 式の演算機能を併せ持つ。

#### 【 0 0 3 5 】

図 8 は 1 組の低域変換係数と高域変換係数を出力するために必要となる最低限の格子点データ演算器を示していて、該構成 (接続関係) はそのままハードウェアに置き換えることが可能である。

#### 【 0 0 3 6 】

各格子点データ演算器は図 6, 図 7 に示すような構成で実現できる。図 6 において、601, 603, 605 は 3 つのデータを入力する端子、607 は演算した格子点データを出力する端子、611 は両端の 2 つの入力データを加算する加算器、613 は該加算結果に係数 ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  の内の 1 つ) を乗算する乗算器、615 は真中の入力データに該乗算結果を加算する加算器である。

#### 【 0 0 3 7 】

乗算器 613 における乗算係数が  $\alpha$  であるものは (1) 式に対応する単機能格

子点データ演算器になり、乗算係数が $\beta$ 、 $\gamma$ 、 $\delta$ であれば、各々(2)(3)(4)式に対応する単機能格子点データ演算器となる。

#### 【0038】

今、乗算係数が $\alpha$ 、端子601、603、605にデータ $X_2$ 、 $X_3$ 、 $X_4$ が入力されているとすると、(1)式にのっとり、 $D_2 = X_2 + \alpha(X_1 + X_3)$ が演算され端子607から出力される。

#### 【0039】

図7では、上記図6の構成に加え、位取り変換器703、セレクタ701、該セレクタの切換用制御信号を入力する端子705が増えている。前記図6の演算機能に加え、境界データの演算機能を有する。

#### 【0040】

端子603に先頭の境界データが入力されていて、端子601には有効なデータが入力されていない時、端子605から入力されたデータを位取り変換器703にて1ビット位を上げて2倍とし、それを前記加算器611の出力の替わりにセレクタ701で選択して以後の計算を行ない、該演算結果を境界処理したデータとして端子607から出力する。この演算は端子603に先頭の境界データが入力された時だけ実行するものであるため、端子705から入力する制御信号でそのタイミングを制御する。

#### 【0041】

図6、図7で説明した格子点データ演算器を図8のように配置すれば、全体としても、問題無く境界処理が行なえる。より理解しやすいように、境界処理をする時の有効なデータの流れを実線で、無効なデータの流れを破線で示したものを図9乃至図11に示す。

図9は、先頭データ $X_0$ がまだ処理対象位置に到達していない状態なので処理出力は得られず、すべての演算は無効となる。よって、全てのデータの流れは破線で示される。該リフティング演算ではあるデータ入力に対して2つの出力データが得られるため、入力データの数と同数の出力データを得るには、入力データを2タップずつ移動すればよい。そのため、以下に説明する図10、11では図

9の入力データを2つつ左に移動した入力状態に対して処理する場合を考える。

#### 【0042】

図10は、先頭データX0がちょうど境界処理の位置にある場合である。先頭データX0より左側には入力データが無いので、左側4つの格子点データ演算器への入出力データは無効となる。この時E0とL0を演算する格子点データ演算器は有効な入力データが2つしか無いので、境界処理に対応した演算モードで処理を行なう。もちろん演算モードを切り換えるための制御信号は入力データのタイミングに併せて0→1→0と切り換える。

#### 【0043】

図11は、先頭データX0が境界処理位置を過ぎた状態である。この場合L2を計算するためにE0が必要であり、該E0を計算する格子点データ演算器には有効な入力データが2つしか無いので、該格子点データ演算器は境界処理に対応した演算モードで処理を行なう必要がある。

#### 【0044】

以上の説明は、先頭の入力データを低域係数へ変換する場合である。先頭の入力データを高域係数に変換する場合には、別の構成で処理をしなければならない。その構成を図12に示す。また、先頭の入力データを高域係数に変換するには、先頭データの入力位置も上述の低域係数へ変換する場合と異なり、1つつらさなければならない。それも併せて図12に示す。

#### 【0045】

先頭データの入力位置を1つつらすのに対応して、境界格子点データ演算器●の位置も1つつ左に移動させている。

#### 【0046】

境界格子点データ演算器●の移動方向は左に限定されるものではなく、右側であってもよい、その場合の構成は図13のようになる。

#### 【0047】

先頭データを、あるブロックのデータでは低域係数に変換し、別のブロックのデータでは高域係数に変換する必要がある場合、前記図8と図13の両方の機能

を有する図 1 4 の構成で処理する必要がある。また、図 1 3 に対応するものとして図 1 5 の構成も考えられる。

#### 【 0 0 4 8 】

以上の如く、画像に対してウェーブレット変換処理することにより、予め境界外のデータを別途用意しなくても、境界部を処理するタイミングにおいて、見かけ上境界で折り返したデータを生成したかのごとく処理することが出来る。従って、フィルタ処理する以前に境界の外側のデータを用意する処理が不要になり、且つ、境界の外側のデータを格納するための記憶領域も不要にすることが可能になる。

#### 【 0 0 4 9 】

なお、上記例では、境界の先頭データの境界処理に着目していたが、最終データの境界処理（以下では終端処理と称す）にも対応した構成について示す。

#### 【 0 0 5 0 】

終端処理への対応には 2 つの方法が考えられるが、その内の 1 つをここで説明し、もう 1 つを変形例として後述する。

#### 【 0 0 5 1 】

本実施形態の重要なポイントを簡単に述べると、全入力データを左右対称に交換して演算部に入力することである。

#### 【 0 0 5 2 】

そのために、入力データを左右対称に交換するためのクロススイッチ 1 2 0 1 を設ける。演算処理には上記例の図 1 4 或いは図 1 5 を用いる。図 1 4 を用いた場合の構成を図 1 6 に示す。

#### 【 0 0 5 3 】

クロススイッチ 1 2 0 1 は左右の入力データを交換して出力するクロスモードと、交換せずに入力したデータをそのまま真下に出力するスルーモードとがある。

#### 【 0 0 5 4 】

先頭データから途中の或るデータまでスルーモードで動作させ、途中でクロスモードに切り換えて、最終データまで入力する。

【 0 0 5 5 】

モード切り換えの前後で処理結果の出力位置が変化するが、その変化の仕方は、演算処理部が図 1 4 の場合と図 1 5 の場合とで異なる。

【 0 0 5 6 】

図 1 7、1 8、1 9 には、図 1 4 で演算処理した場合の変化の仕方を示し、図 2 0、2 1 には、図 1 5 で演算処理した場合の変化の仕方を示す。

【 0 0 5 7 】

切り換え前は、演算処理した 2 つの係数が図 1 7 のように出力されていたとする。

【 0 0 5 8 】

入力データはそのままクロススイッチ 1 2 0 1 のみモード切り換えをすると、新たな高域係数が図 1 8 のように出力されるが、その後入力データを 2 つずつ移動させると、低域と高域の 2 つの係数は図 1 9 のように出力される。

【 0 0 5 9 】

図 1 7 と図 1 9 は一見すると、同じ出力状態のように見えるが、図 1 7 では左側の高域係数の添え時番号が右側の低域係数の添え時番号よりも小さいが、図 1 9 は入力データが左右で入れ替わっているため、添え時番号の大小関係が反対になる。

【 0 0 6 0 】

この添え時番号の大小関係の逆転は、図 2 0 から図 2 1 に変わる場合にも言える。すなわち、モード切り換え前は左側出力の添え時番号の方が小さいが、モード切り換え後は左側出力の添え時番号の方が大きくなる。

【 0 0 6 1 】

図 2 0、2 1 では、モードを切り換える時に、入力データを一時的にその状態で保つと言った必要が無い。その替わりに出力の位置が変わる。

【 0 0 6 2 】

<変形例>

ここでは、終端処理へ対応するもう 1 つの方法について説明する。

【 0 0 6 3 】



本変形例では前記クロススイッチを使わずに、終端処理用の境界格子点データ演算器と、先頭データと最終データの両方の境界処理が可能な境界格子点データ演算器とを使用する。それぞれの構成を図 2 2 と図 2 3 に示す。

#### 【 0 0 6 4 】

図 7 の先頭データ処理用の境界格子点データ演算器と図 2 2 との違いは、位取り変換器 7 0 3 への入力端子が端子 6 0 5 から端子 6 0 1 へ変更になったことである。

#### 【 0 0 6 5 】

図 2 3 の境界格子点データ演算器は、両方の境界処理をするため、端子 6 0 1 と端子 6 0 5 の入力信号を切り換えて位取り変換器 7 0 3 へ入力できるようになっている。該切り換えはセレクタ 1 6 0 1 にて行ない、該セレクタの切り換え制御信号は端子 1 6 0 3 から入力する。その他の構成要素は前記図 7 とまったく同じである。

#### 【 0 0 6 6 】

図 2 2 と図 2 3 の境界格子点データ演算器をそれぞれ◆と◎で表わし、変換処理部全体の構成を図 2 4 に示す。これが本変形例の構成である。同図において X 25 が最終データを表わし、その時の有効データの流れを実線で、無効なデータの流れを破線で示している。

#### 【 0 0 6 7 】

先頭データを高域係数へ変換する場合、該係数は H 23 の格子点データ演算器から出力され、最終データを高域係数へ変換する場合、該係数は H 25 の格子点データ演算器から出力されるため、低域係数を含めると、3 つの演算器から出力を取り出すことになる。

#### 【 0 0 6 8 】

出力を取り出す演算器を 2 つに限定したい場合には、変換処理部を図 2 5 に示す構成にすればよい。この構成にすれば、すべての高域係数を H 25、低域係数を L 24 の格子点データ演算器から取り出すことができる。

#### 【 0 0 6 9 】

これまでに説明した実施形態及びその変形例の構成はすべて順方向の変換処理

部のみであったが、図 3 の順方向変換処理部と図 4 の逆方向変換処理部との類似性から、実施形態の構成は逆方向変換処理部にも容易に適用できるのは、上記の説明からすれば当業者には容易に想到し得よう。従って、本発明は上記実施形態に限定されるものではない。

## 【 0 0 7 0 】

以上説明したように上記実施形態によれば、格子点データを演算する機能を有する格子点データ演算器群の一部に、境界の内側のデータのみを選択する手段を設け、該選択手段の出力に対して所定の係数を乗算する構成とした。これにより、あらかじめ境界の外側に内側のデータを折り返して作成するといった処理を不要とし、外側のデータを格納するための記憶領域も要らなくなった。

## 【 0 0 7 1 】

## &lt; 第 2 の実施形態 &gt;

本実施形態では、図 2 6 に示す格子点データ演算ユニットを図 2 7 のように多段階接続する構成で、フィルタ処理のリフティング演算を行なう方式である。

## 【 0 0 7 2 】

図 2 6 において、2 6 0 1, 2 6 0 3 は 2 つのデータを入力する端子、2 6 0 7 は演算した格子点データを出力する端子、2 6 2 1 は端子 2 6 0 3 からの入力データを格納するバッファ、2 6 0 9 はバッファ 2 6 2 1 の出力を外部へ出力する端子、2 6 1 1 はバッファ 2 6 2 1 の出力データと入力データを加算する加算器、2 6 1 3 は該加算結果に係数 ( $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  の内の 1 つ) を乗算する乗算器、2 6 1 5 は演算に用いる 3 つのデータの真中に位置する入力データに該乗算結果を加算する加算器である。

## 【 0 0 7 3 】

以下では、図 3 を再度参考にして、該演算方式の概要を簡単に説明する。

## 【 0 0 7 4 】

まず、9 個の入力データ  $X_0$ ,  $X_1$ ,  $X_2$ ,  $X_3$ ,  $X_4$ ,  $X_5$ ,  $X_6$ ,  $X_7$ ,  $X_8$  を処理する場合について考える。この場合、10 個の格子点データ ( $D_1$ ,  $D_3$ ,  $D_5$ ,  $D_7$ ,  $E_2$ ,  $E_4$ ,  $E_6$ ,  $H_3$ ,  $H_5$ ,  $L_4$ ) を演算することにより、低域変換係数  $L_4$  と高域変換係数  $H_5$  を出力することができる。

## 【 0 0 7 5 】

次に X 9, X 1 0 の 2 つデータが新たに入力データとして加わると、同様に 1 0 個の格子点データを演算することにより、低域変換係数 L 6 と高域変換係数 H 7 を出力することができるが、この時、以前に演算した格子点データを利用することが出来れば、新たに計算しなければならないのは、D 9, E 8, L 7, L 6 の 4 つだけですむ。

## 【 0 0 7 6 】

以前に演算した格子点データを利用するには該格子点データを記憶し保持するための手段が必要であり、それが図 2 6 におけるバッファ 2 6 2 1 である。

## 【 0 0 7 7 】

図 2 7 における最上段の格子点データ演算ユニット 2 7 0 1 内のバッファのみは、以前に演算した格子点データでは無く、以前に入力したデータを保持するために用いられるが、その他の格子点データ演算ユニット内のバッファは以前演算した格子点データを保持するために用いる。このバッファのサイズは最小 1 で上限は無い。

## 【 0 0 7 8 】

具体的に、低域変換係数 L 6 と高域変換係数 H 7 を出力する場合について考察してみる。最上段の格子点データ演算ユニット 2 7 0 1 には、新たな入力データ X 9, X 1 0 の 2 つのみが入力される。該格子点データ演算ユニットでは、D 9 を演算するが、該演算に必要なデータ X 8 は図 2 6 のバッファ 2 6 2 1 から出力される。この X 8 は以前 X 8 が端子 2 6 0 3 入力から入力された時に該バッファは格納されたものである。

## 【 0 0 7 9 】

格子点データ演算ユニット 2 7 0 1 は、演算した D 9 とバッファからの出力 X 8 をそれぞれ端子 2 6 0 7 と 2 6 0 9 からユニットの外部へ出力し、次の格子点データ演算ユニット 2 7 0 2 に入力する。

## 【 0 0 8 0 】

格子点データ演算ユニット 2 7 0 2 は、上記入力に基づき E 8 を演算するが、該演算に必要な他のデータ D 7 は該ユニット内のバッファ 2 6 2 1 から出力され

る。このD 7も以前端子2 6 0 3から入力された時に該バッファに格納したものである。演算したE 8とバッファからの出力D 7をそれぞれ端子2 6 0 7、2 6 0 9からユニットの外へ出力し、次の格子点データ演算ユニット2 7 0 3に送る。

#### 【0 0 8 1】

格子点データ演算ユニット2 7 0 3、2 7 0 4も同様の処理を行なう。その結果、演算ユニット2 7 0 3からは高域変換係数H 7が、演算ユニット2 7 0 4からは低域変換係数L 6が出力される。

#### 【0 0 8 2】

これ以降、演算ユニット2 7 0 1に新たなデータが2つ入力される毎に、演算ユニット2 7 0 3と2 7 0 4から高域と低域め変換係数が出力される。

#### 【0 0 8 3】

本第2の実施形態では、ウェーブレット変換あるいはフィルタ演算処理する際に、画像境界部の内側のデータを外側に折り返して用いる境界処理と等価な処理を、効率よく行なうための各種（格子点データ）演算ユニットの構成と、それらの演算ユニットをどのように接続してフィルタ処理をするのかを開示するものである。

#### 【0 0 8 4】

新たに導入する演算ユニットについては、先に説明した式（1 a）乃至（8 b）をそのまま用いるものとする。式（1 a）～（4 a），（5 a）～（8 a）は先頭の境界データに対応する演算内容で、（1 b）～（4 b），（5 b）～（8 b）は最後の境界データに対応する演算内容である。

#### 【0 0 8 5】

図2 7の格子点データ演算ユニット2 7 0 1～2 7 0 4は先に示した式（1）～（4）式の演算機能を有しているが、それに加えさらに式（1 a）～（4 a）式、あるいは（1 b）～（4 b）式の演算機能を有する演算ユニットを考える。

#### 【0 0 8 6】

式（1）と（1 a）式の2つの演算機能、あるいは、式（2）と（2 a）、式

(3) と (3 a)、式 (4) と (4 a) の 2 つの演算機能を有する演算ユニットを始端対応 (格子点データ) 演算ユニットと称し、その構成は図 2 8 に示す通りである。

## 【 0 0 8 7 】

図 2 8 では、上記図 2 6 の構成に加え、位取り変換器 2 8 0 1、セレクタ 2 8 0 3、該セレクタの切換用制御信号を入力する端子 2 8 0 5 が増えている。先頭の境界データを処理するタイミングで、前記制御信号を“1”にして、その他では“0.”にすることにより、先頭の境界データは (1 a) 式で処理し、その他のデータは (1) 式で処理することができる。

## 【 0 0 8 8 】

乗算器 2 6 1 3 における乗算係数が  $\alpha$  であるものは (1) と (1 a) 式を演算する始端対応演算ユニットになり、乗算係数が  $\beta$ 、 $\gamma$ 、 $\delta$  であれば、各々 (2) と (2 a)、(3) と (3 a)、(4) と (4 a) 式に対応する始端対応演算ユニットとなる。

## 【 0 0 8 9 】

同様に (1) と (1 b) 式の 2 つの演算機能、あるいは、(2) と (2 b) 式、(3) と (3 b) 式、(4) と (4 b) 式の 2 つの演算機能を有する演算ユニットを終端対応 (格子点データ) 演算ユニットと称する。この構成は図 2 9 に示す如くになる。基本的な構成は図 2 8 と同じであるが、1 つだけ違いがある。それは位取り変換器 2 8 0 1 への入力信号がバッファ 2 6 2 1 の出力に変わったことである。この変更により上記 (1 b) ~ (4 b) 式の処理、すなわち最終データに対する境界処理が可能になる。

## 【 0 0 9 0 】

本第 2 の実施形態では、上述した始端対応演算ユニットと終端対応演算ユニットを用いて、処理するデータのサイズ (連続するデータの数) が偶数なら、両端のデータをきちんと境界処理することができる構成を示す。図 3 0 及び図 3 4 に示した構成がそれである。

## 【 0 0 9 1 】

図 3 0 の構成は先頭データを低域係数に変換する場合に対応するもので、図 3

4 は先頭データを高域係数に変換する場合に対応する。この 2 つの構成の違いは始端対応演算ユニットと終端対応演算ユニットとが入れ替わっていることである。

#### 【 0 0 9 2 】

また、最上段の演算ユニットに先頭データ (X 0) を入力する端子も違っている。図 3 0 では端子 2 6 0 3 に、図 3 4 では端子 2 6 0 1 に先頭データを入力する。

#### 【 0 0 9 3 】

以下では図 3 0 の構成の動作について、図 3 1 と図 3 2 を参照して説明する。図 3 1 はフィルタ処理全体の概略を、図 3 2 は各サイクルにおける各演算ユニットの入出力を表わしている。

#### 【 0 0 9 4 】

図 3 1 は先頭データを低域係数に変換する場合において、計算しなければならない格子点データを示しており、特に●で示した格子点データは上述の境界処理用の演算で求める必要がある。具体的に、E 0 は (2 a) 式、L 0 は (4 a) 式、D 1 1 は (1 b) 式、H 1 1 は (3 b) 式で演算して求める。

#### 【 0 0 9 5 】

サイクル A では、演算ユニット 3 0 0 1 の入力端子 2 6 0 3 に X 0 だけを入力する。この入力データ X 0 はバッファ 2 6 2 1 に格納される。この時、他の演算ユニットは他のブロックのデータを処理しているか何も処理していない。図 3 2 では着目したデータを処理している演算ユニットを実線で、他の演算ユニットを破線で表わすことで、演算ユニットを区別している。

#### 【 0 0 9 6 】

サイクル B では、同じ入力端子 2 6 0 3 に X 2 を、入力端子 2 6 0 1 に X 1 を入力する。演算ユニット 3 0 0 1 はバッファ出力の X 0 と上記入力データ X 1, X 2 の 3 つのデータから D 1 を演算し、該演算結果 D 1 とバッファ出力の X 0 を次の演算ユニット 3 0 0 2 へ出力する。これより、演算ユニット 3 0 0 1 は始端対応演算ユニットでなくてよいことが解かる。

#### 【 0 0 9 7 】

演算ユニット 3 0 0 2 は、上記入力データ X 0, D 1 の 2 つのデータから E 0 を演算し、該演算結果 E 0 のみを次の演算ユニット 3 0 0 3 の端子 2 6 0 3 へ出力する。この時、該演算ユニットは前記 (2 a) 式に基づいて始端境界処理を行なう。よって、演算ユニット 3 0 0 2 は始端対応演算ユニットでなければならない。

【 0 0 9 8 】

演算ユニット 3 0 0 3 は、上記入力データ E 0 をバッファ 2 6 2 1 に格納する。

【 0 0 9 9 】

サイクル C では、入力端子 2 6 0 1 に X 3 を、入力端子 2 6 0 3 に X 4 を入力する。演算ユニット 3 0 0 1 はバッファ出力の X 2 と上記入力データ X 3, X 4 の 3 つのデータから D 3 を演算し、該演算結果 D 3 とバッファ出力の X 2 を次の演算ユニット 3 0 0 2 へ入力する。

【 0 1 0 0 】

演算ユニット 3 0 0 2 は、バッファ出力の D 1 と上記入力データ X 2, D 3 の 3 つのデータから E 2 を演算し、該演算結果 E 2 とバッファ出力の D 1 を次の演算ユニット 3 0 0 3 へ出力する。

【 0 1 0 1 】

演算ユニット 3 0 0 3 は、バッファ出力の E 0 と上記入力データ D 1, E 2 の 3 つのデータから H 1 を演算し、該演算結果 H 1 とバッファ出力の E 0 を次の演算ユニット 3 0 0 4 へ出力する。

【 0 1 0 2 】

演算ユニット 3 0 0 4 は E 0, H 1 の 2 つのデータから L 0 を演算し、該演算結果 L 0 のみ出力する。この時、該演算ユニットは前記 (4 a) 式に基づいて始端境界処理を行なう。よって、演算ユニット 3 0 0 4 は始端対応演算ユニットでなければならない。

【 0 1 0 3 】

サイクル A、サイクル B、サイクル C は時間的に連続する必要は無いが、各演算ユニットは入力データとバッファから読み出すデータとが関連するように制御

する必要がある。最も簡単にそれを実現する方法は、周期的に関連するデータを処理することである。

#### 【 0 1 0 4 】

以上は、先頭データの境界処理の説明であった。次に、最終データの境界処理について図 3 1 と図 3 3 を参照して説明する。参照する図 3 1 に対応して、最終データを X 1 1 とする。

#### 【 0 1 0 5 】

サイクル P に、最終データ X 1 1 が演算ユニット 3 0 0 1 の入力端子 2 6 0 1 に入力されたとする。この時、演算ユニット 3 0 0 1 ～ 3 0 0 4 のバッファには、X 1 0, D 9, E 8, H 7 が格納されている。

#### 【 0 1 0 6 】

演算ユニット 3 0 0 1 は、バッファの出力の X 1 0 と上記入力データ X 1 1 の 2 つのデータから D 1 1 を演算し、該演算結果 D 1 1 とバッファ 2 6 2 1 の出力 X 1 0 を、次の演算ユニット 3 0 0 2 に出力する。この時、該演算ユニットは前記 ( 1 b ) 式に基づいて終端境界処理を行なう。よって、演算ユニット 3 0 0 1 は終端対応演算ユニットでなければならない。

#### 【 0 1 0 7 】

演算ユニット 3 0 0 2 は、バッファの出力の D 9 と上記入力データ X 1 0, D 1 1 の 3 つのデータから E 1 0 を演算し、該演算結果 E 1 0 とバッファ出力の D 9 を次の演算ユニット 3 1 0 0 3 に出力する。この時、入力データ D 1 1 をバッファに格納する。

#### 【 0 1 0 8 】

演算ユニット 3 0 0 3 は、バッファの出力 E 8 と上記入力データ D 9, E 1 0 の 3 つのデータから H 9 を演算し、該演算結果 H 9 とバッファ出力の E 8 を次の演算ユニット 3 0 0 4 に入力する。この時、入力データ E 1 0 をバッファに格納する。

#### 【 0 1 0 9 】

演算ユニット 3 0 0 4 は、バッファの出力 H 7 と上記入力データ E 8, H 9 の 3 つのデータから L 8 を演算し、該演算結果 L 8 をフィルタ処理結果として出力



する。

【 0 1 1 0 】

上記各演算ユニットの入出力データの内容は図 3 3 < サイクル P > にも示してある。

【 0 1 1 1 】

サイクル Q では、演算ユニット 3 0 0 2 内のバッファ出力 D 1 1 を、次の演算ユニット 3 0 0 3 に出力する。

【 0 1 1 2 】

演算ユニット 3 0 0 3 は、バッファの出力 E 1 0 と上記入力データ D 1 1 の 2 つのデータから H 1 1 を演算し、該演算結果 H 1 1 とバッファ出力の E 1 0 を次の演算ユニット 3 0 0 4 に出力する。この時、該演算ユニットは前記 ( 3 b ) 式に基づいて終端境界処理を行なう。よって、演算ユニット 3 0 0 3 は終端対応演算ユニットでなければならないことが解かる。演算結果 H 1 1 はフィルタ処理結果として外部にも出力される。

【 0 1 1 3 】

演算ユニット 3 0 0 4 は、バッファの出力 H 9 と上記入力データ E 1 0 , H 1 1 の 3 つのデータから L 1 0 を演算し、該演算結果 L 1 0 をフィルタ処理結果として出力する。先頭データの境界処理の説明と同様、サイクル P とサイクル Q は連続する必要は無い。

【 0 1 1 4 】

以上を整理すると、演算ユニット 3 0 0 2 と 3 0 0 4 は始端対応演算ユニットでなければならない、演算ユニット 3 0 0 1 と 3 0 0 3 は終端対応演算ユニットでなければならない。

【 0 1 1 5 】

そして、各始端対応ユニットが境界処理 ( セレクタ切り換え ) をするのは先頭データ ( 上記説明では添え字が 0 のデータ ) が入力される 1 サイクルのみであり、各終端対応ユニットが境界処理 ( セレクタ切り換え ) をするのは最終データ ( 上記説明では添え字が 1 1 のデータ ) が入力される 1 サイクルのみである。

【 0 1 1 6 】

先頭データを低域係数へ変換する場合の構成である図 3 0 の説明は以上で終わり、次に先頭データを高域係数へ変換する場合の構成である図 3 4 の説明に移る。

#### 【 0 1 1 7 】

ここでは、先頭データを高域係数へ変換する場合のLifting格子構造を図 3 5 に示し、それを用いて説明する。

#### 【 0 1 1 8 】

図 3 5 を図 3 1 と比較すると、境界処理演算で求める●の格子点データの位置が左右で逆の関係になっていることが解かる。これは始端対応演算ユニットと終端対応ユニットを入れ替えることに対応している。

#### 【 0 1 1 9 】

すなわち、図 3 0 の始端対応演算ユニットと終端対応ユニットを入れ替えた図 3 4 の構成で図 3 5 の各格子点データを演算でき、先頭データを高域係数に変換することができる。

#### 【 0 1 2 0 】

本第 2 の実施形態の構成では、前のブロックの最終データと次のブロックの先頭データを連結して連続処理しても、それぞれの境界処理を独立に行なうことができ、2つのブロックのデータを干渉させることはない。

#### 【 0 1 2 1 】

ブロックの境界に現われる最終データや先頭データなどの境界データ、或いは、該境界データと同位置（同じ添え時番号）の変換データを各格子点データ演算ユニットに入力するタイミングで、各演算ユニットが入力データに対応した境界処理を行なうことで連続処理は実現される。入力データに対応した境界処理機能を有しない時は通常の演算処理を行なう。

#### 【 0 1 2 2 】

これにより、サイズの大きいデータを複数のブロックに分割して、各ブロックの境界に上記境界処理を施す場合でも、該サイズの大きいデータを連続的に処理することができるようになる。

#### 【 0 1 2 3 】

### ＜第 3 の実施形態＞

前記第 2 の実施形態では、処理するデータのサイズを偶数に限定していた。本第 3 の実施形態では、処理するデータのサイズが奇数の時に、先頭データを低域係数へ変換する場合と高域係数へ変換する場合の構成について示す。

#### 【 0 1 2 4 】

本第 3 の実施形態は、図 3 6 に示す構成の演算ユニットを用いる。該演算ユニットは、前記始端対応演算ユニットと終端対応演算ユニットの両方の機能を有する。そこで該演算ユニットを両端対応演算ユニットと称す。前記両方の機能はセレクタ 3 6 0 1 を設けることにより実現した。該セレクタへの制御信号は端子 3 6 0 3 より入力する。

#### 【 0 1 2 5 】

両端対応演算ユニットを用いて、奇数サイズのデータの先頭データを低域係数へ変換する場合の構成を図 3 7 に、高域係数へ変換する場合の構成を図 3 8 に示す。

#### 【 0 1 2 6 】

リフティング演算でフィルタ処理する時に求める格子点データの中で、境界処理が必要な格子点データの数は、全部でリフティング演算の段数と同じ数になる。図 1 のようにリフティング演算の段数が 4 段である時は、先頭データと最終データに対してそれぞれ 2 つ格子点データに境界処理が必要である。

#### 【 0 1 2 7 】

上記の第 2 の実施形態のようにデータサイズが偶数の場合、始端境界処理が必要なりフティング演算と終端境界処理が必要なりフティング演算は交互に位置し重なることは無かった（図 3 1、図 3 5 を参照）。

#### 【 0 1 2 8 】

しかし、データサイズが奇数の場合、始端境界処理が必要なりフティング演算と終端境界処理が必要なりフティング演算は完全に一致する（図 3 9、図 4 0 参照）。

#### 【 0 1 2 9 】

先頭データを低域係数へ変換する場合には図 3 9 に示すように先頭データと最

終データの両方において、2 段目と 4 段目のリフティング演算で境界処理を行なう必要がある。

#### 【 0 1 3 0 】

先頭データを高域係数へ変換する場合には図 4 0 に示すように先頭データと最終データの両方において、1 段目と 3 段目のリフティング演算で境界処理を行なう必要がある。

#### 【 0 1 3 1 】

これは、一部の格子点データ演算ユニットのみに、始端と終端の両方の境界処理機能が要求されることを意味する。

#### 【 0 1 3 2 】

すなわち、上記一部の格子点データ演算ユニットを、図 3 6 に示す両端対応演算ユニットに置き換えることで奇数サイズのデータの境界処理が可能になる。

#### 【 0 1 3 3 】

図 3 7 の構成は、2 段目と 4 段目の格子点データ演算ユニットを図 3 6 に示す両端対応演算ユニットに置き換えたものであり、1 段目と 3 段目は前記図 2 7 における格子点データ演算ユニットと同じである。この構成により、図 3 9 に対応した処理が可能である。

#### 【 0 1 3 4 】

図 3 8 の構成は、1 段目と 3 段目の格子点データ演算ユニットを図 3 6 に示す両端対応演算ユニットに置き換えたものであり、2 段目と 4 段目は前記図 2 7 における格子点データ演算ユニットと同じである。この構成により、図 4 0 に対応した処理が可能である。

#### 【 0 1 3 5 】

本実施形態の構成では、前のブロックの最終データと次のブロックの先頭データとの間にダミーデータを 1 つ入れることで、複数のブロックを連続処理することが可能である。

#### 【 0 1 3 6 】

ダミーデータが無いと、前ブロックの最終データの変換係数と次ブロックの先頭データの変換係数とが違う種類の係数になってしまい、境界データを低域係数

あるいは高域係数に統一して変換するといったことが出来なくなってしまう。

#### 【0 1 3 7】

##### ＜第4の実施形態＞

すべての格子点データ演算ユニットを両端対応演算ユニットに置き換えた図4 1の構成なら、境界データを低域と高域係数のいずれに変換してもうまく境界処理できるだけでなく、第2の実施形態の機能を包含するため、データサイズが偶数と奇数のいずれであってもうまく境界処理ができる。図4 1には1つの処理例として、先頭データを低域係数へ変換する場合の入出力データの例が示してある。

#### 【0 1 3 8】

さらに、図3 6の両端対応演算ユニットを図4 2のようにわずかに変更を加える。変更内容は、図3 6における加算器2 6 1 5を加減算器4 2 0 1にただけである。これは図2の逆方向のリフティング演算に対応するためである（変更がわずかなので名称は同じ両端対応演算ユニットとする）。上記変更に加え、図4 3のように各々の両端対応演算ユニット間にセレクタを配し、セレクタ4 3 1 4→演算ユニット4 3 0 4→セレクタ4 3 1 3→演算ユニット4 3 0 3→セレクタ4 3 1 2→演算ユニット4 3 0 2→セレクタ4 3 1 1→演算ユニット4 3 0 1のように逆に処理することにより、逆変換処理も可能となる。

#### 【0 1 3 9】

順方向の変換は、セレクタ4 3 1 1→演算ユニット4 3 0 1→セレクタ4 3 1 2→演算ユニット4 3 0 2→セレクタ4 3 1 3→演算ユニット4 3 0 3→セレクタ4 3 1 4→演算ユニット4 3 0 4の順に処理する（逆変換時の入出力データには頭に\*印をつけて区別してある。）。

#### 【0 1 4 0】

順方向変換と逆方向変換の切り換えは、端子4 3 2 1から入力する制御信号により行なわれる。該制御信号によりセレクタ4 3 1 1～4 3 1 4と演算ユニット内の加減算器4 2 0 1の演算機能を切り換えることにより、上記のような2通りの演算順序が実現できる。該加減算器4 2 0 1は、順方向変換時には加算器として動作し、逆変換時には減算器として動作する。

## 【 0 1 4 1 】

以上説明したように本第 2 乃至第 4 の実施形態では、入力データを格納するバッファと格子点データを演算する演算部を有する格子点データ演算ユニット内に、境界の内側のデータのみを選択する手段を設け、該選択手段の出力に対して所定の係数を乗算する構成とした。これにより、あらかじめ境界の外側に内側のデータを折り返して作成するといった処理を不要とし、外側のデータを格納するための記憶領域も要らなくなった。

## 【 0 1 4 2 】

## 【発明の効果】

以上説明したように本発明によれば、処理に必要なデータが画像データの境界外にある場合であっても、その境界外のデータを前処理で生成する必要が無く、以って構造を簡略化させることが可能になる。

## 【図面の簡単な説明】

## 【図 1】

順方向のLifting Schemeを示す図である。

## 【図 2】

逆方向のLifting Schemeを示す図である。

## 【図 3】

順変換のリフティング格子構造を示す図である。

## 【図 4】

逆変換のリフティング格子構造を示す図である。

## 【図 5】

境界データの外側における入力データの折り返しを示す図である。

## 【図 6】

格子点データ演算ユニットの構成を示す図である。

## 【図 7】

上記格子点データ演算ユニットを用いて構成したウェーブレット変換処理部を示す図である。

## 【図 8】

第 1 の実施形態の構成を示す図である。

【図 9】

第 1 の実施形態の動作を説明するための図である。

【図 1 0】

第 1 の実施形態の動作を説明するための図である。

【図 1 1】

第 1 の実施形態の動作を説明するための図である。

【図 1 2】

第 1 の実施形態の他の構成を示す図である。

【図 1 3】

第 1 の実施形態の他の構成を示す図である。

【図 1 4】

第 1 の実施形態の構成を拡張した図である。

【図 1 5】

第 1 の実施形態の構成を拡張した図である。

【図 1 6】

第 1 の実施形態の変形例の構成を示す図である。

【図 1 7】

第 1 の実施形態の変形例の動作を説明するための図である。

【図 1 8】

第 1 の実施形態の変形例の動作を説明するための図である。

【図 1 9】

第 1 の実施形態の変形例の動作を説明するための図である。

【図 2 0】

第 1 の実施形態の他の構成の動作を説明するための図である。

【図 2 1】

第 1 の実施形態の他の構成の動作を説明するための図である。

【図 2 2】

第 1 の実施形態の変形例で用いる終端処理用の格子点データ演算器の構成を示

す図である。

【図 2 3】

第 1 の実施形態の変形例で用いる終端処理と終端処理の両方が可能な格子点データ演算器の構成を示す図である。

【図 2 4】

第 1 の実施形態の変形例の構成を示す図である。

【図 2 5】

第 1 の実施形態の変形例の構成を示す図である。

【図 2 6】

第 2 の実施形態における格子点データ演算ユニットの構成を示す図である。

【図 2 7】

第 2 の実施形態における格子点データ演算ユニットを用いて構成したフィルタ処理部のブロック構成図である。

【図 2 8】

第 2 の実施形態における始端対応演算ユニットの構成を示す図である。

【図 2 9】

第 2 の実施形態における終端対応演算ユニットの構成を示す図である。

【図 3 0】

第 2 の実施形態におけるフィルタ処理装置のブロック構成図である。

【図 3 1】

データサイズが偶数の場合に境界処理が必要な演算を示す図である。

【図 3 2】

第 2 の実施形態における始端境界処理のタイミングを示す図である。

【図 3 3】

第 2 の実施形態における終端境界処理のタイミングを示す図である。

【図 3 4】

第 2 の実施形態の他の構成を示す図である。

【図 3 5】

データサイズが偶数の場合に境界処理が必要な演算を示す図である。



【図 3 6】

第 3 の実施形態における両端対応演算ユニットの構成を示す図である。

【図 3 7】

第 3 の実施形態におけるフィルタ処理装置のブロック構成図である。

【図 3 8】

第 3 の実施形態におけるフィルタ処理装置の他のブロック構成図である。

【図 3 9】

データサイズが奇数の場合に境界処理が必要な演算を示す図である。

【図 4 0】

データサイズが偶数の場合に境界処理が必要な演算を示す図である。

【図 4 1】

第 4 の実施形態におけるフィルタ処理装置のブロック構成図である。

【図 4 2】

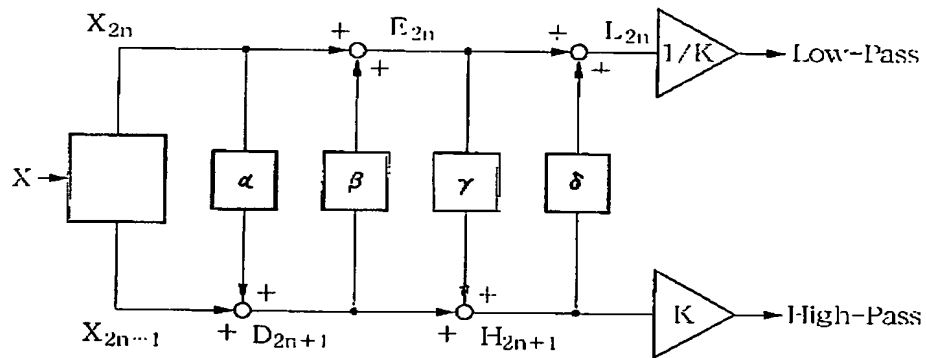
第 4 の実施形態における演算ユニットの構成を示す図である。

【図 4 3】

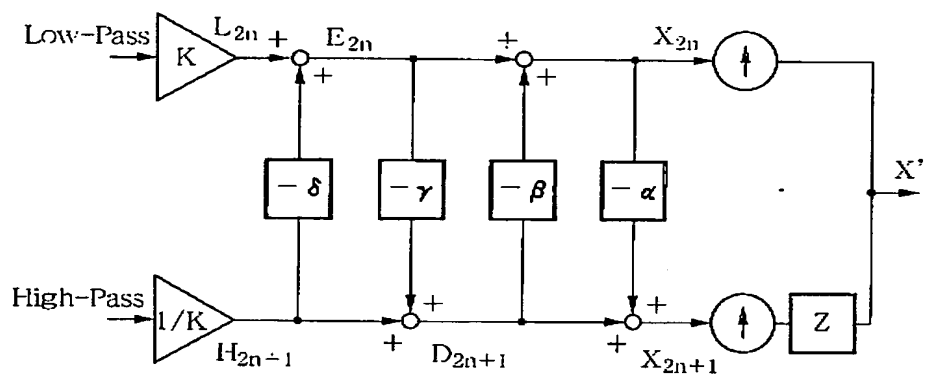
第 4 の実施形態におけるフィルタ処理装置の他のブロック構成図である。

【書類名】 図面

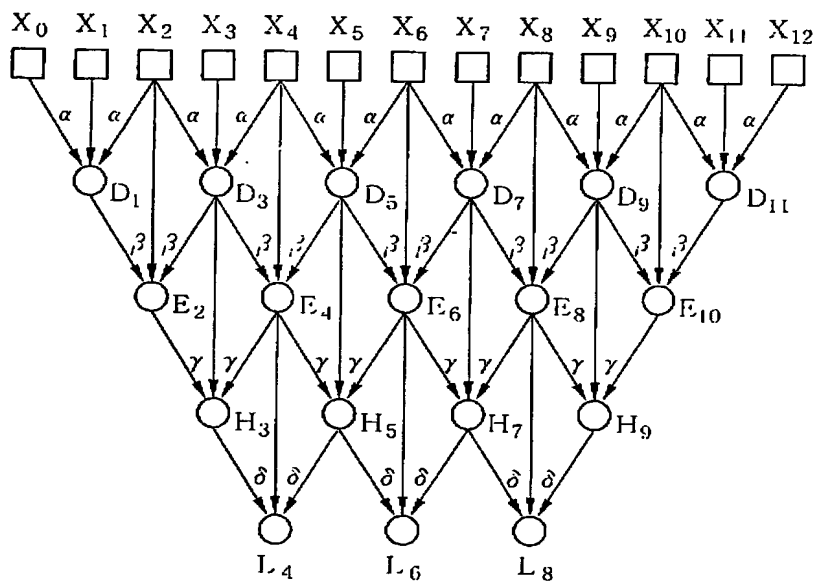
【図 1】



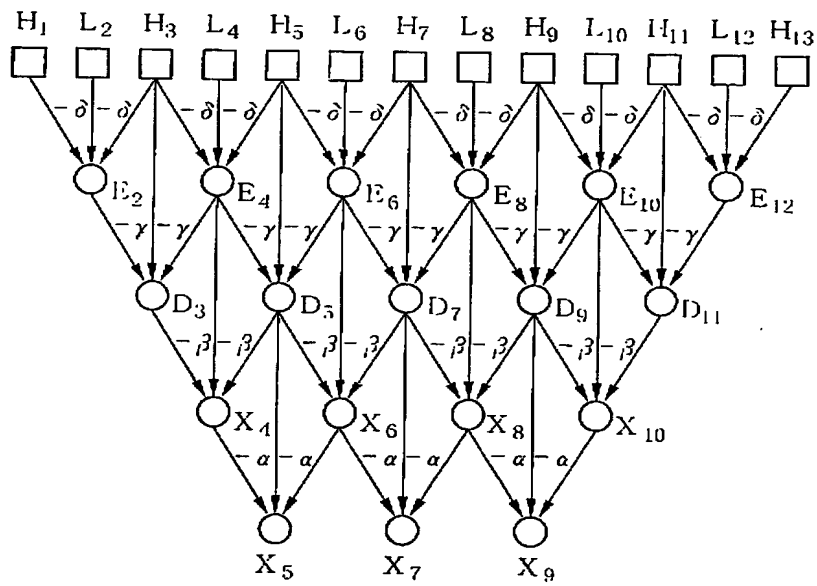
【図 2】



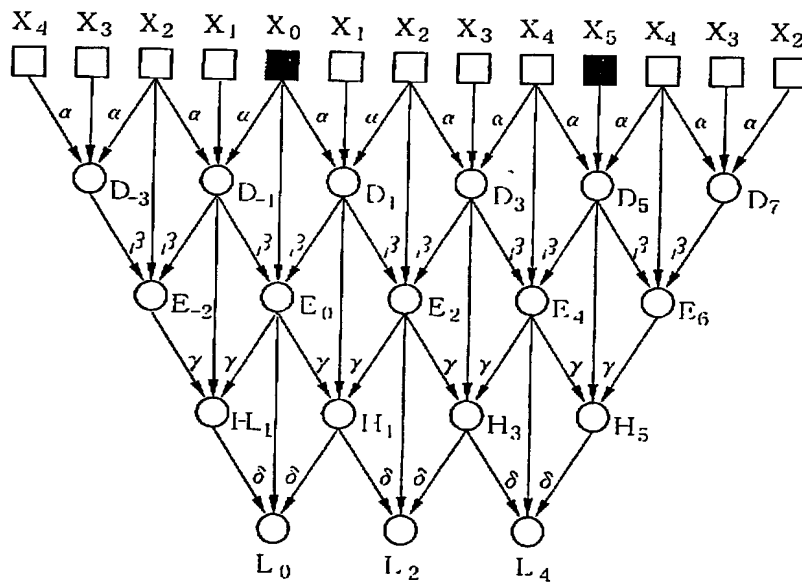
【図 3】



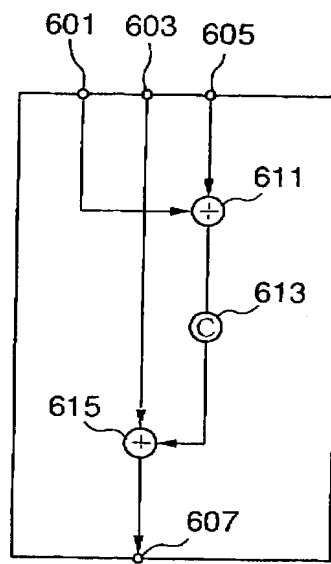
【図 4】



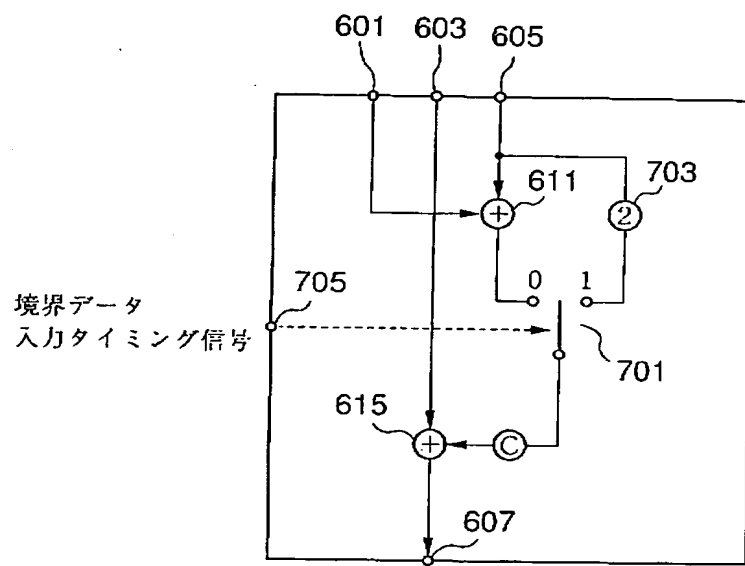
【図 5】



【図 6】

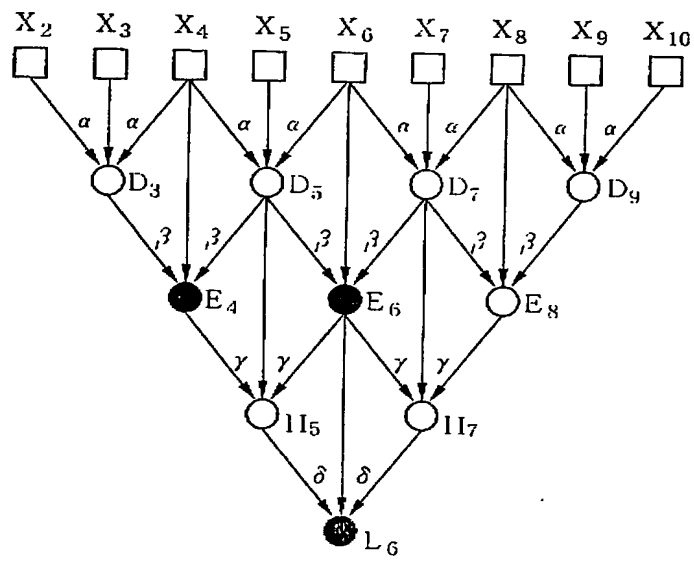


【図 7】

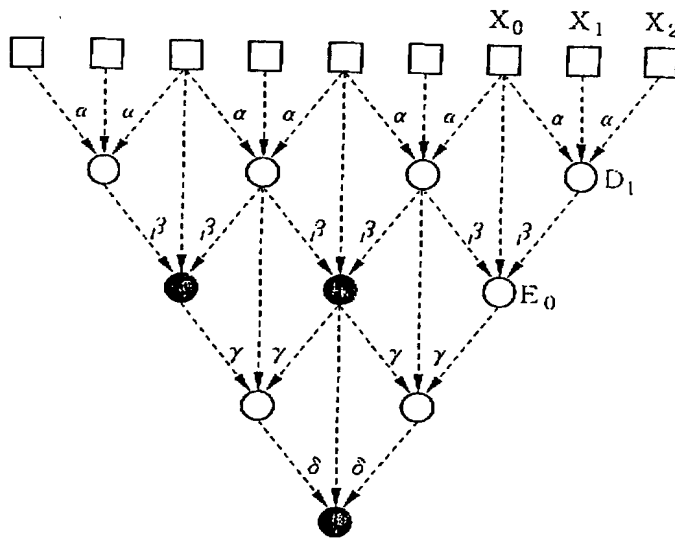




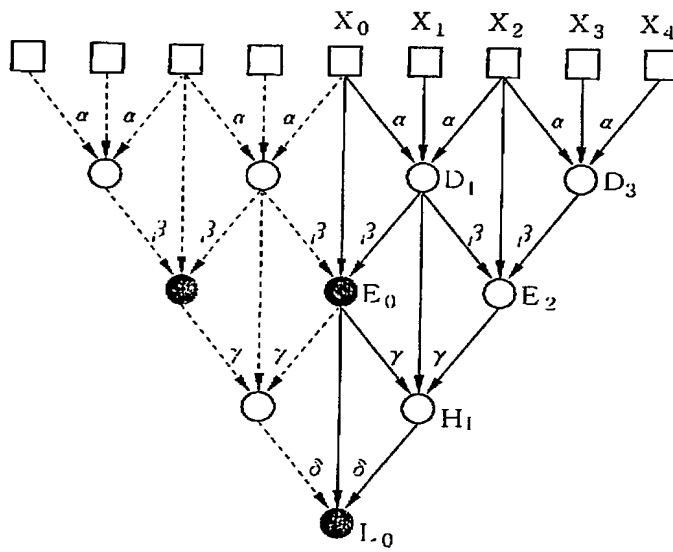
【図 8】



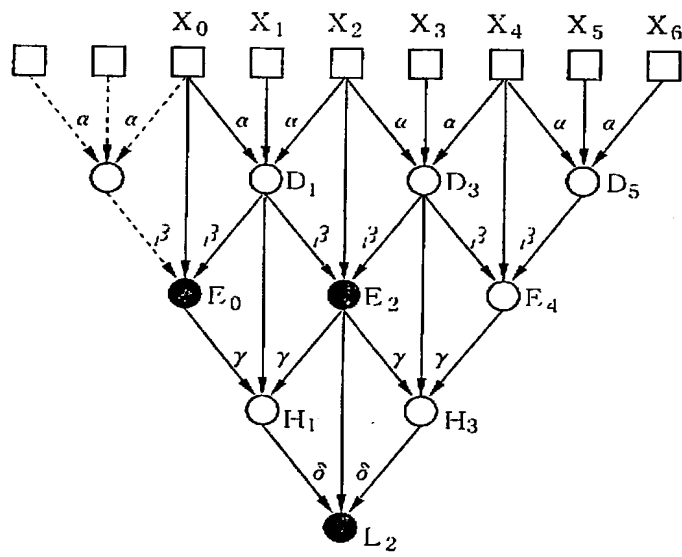
【図 9】



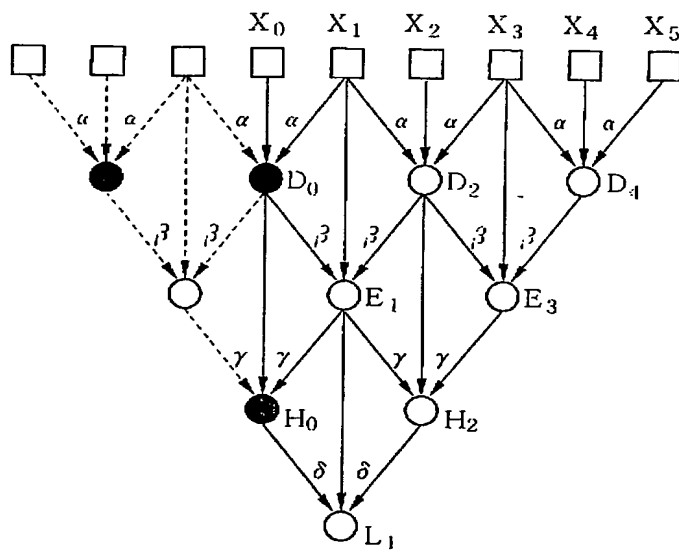
【図 1 0】



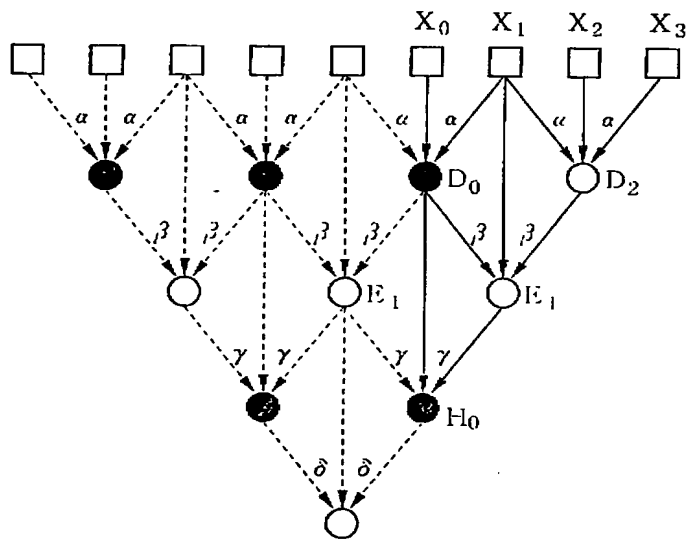
【図 1 1】



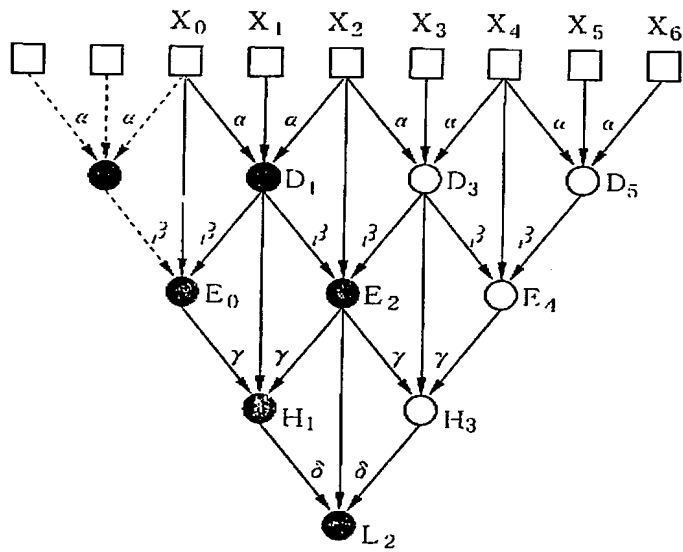
【図 1 2】



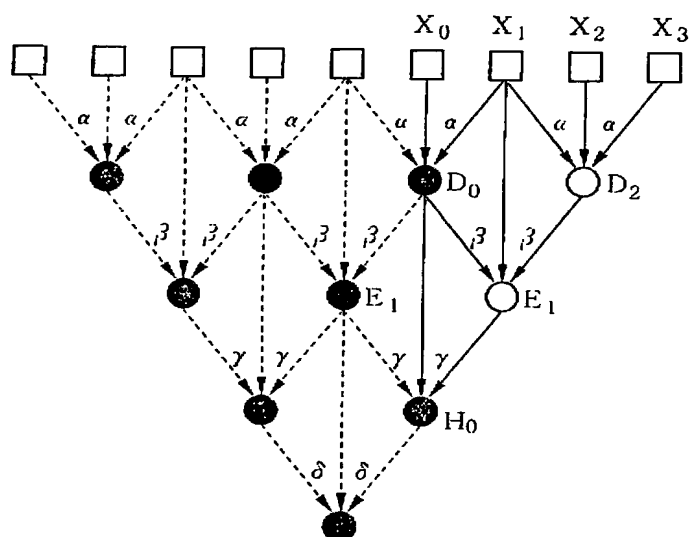
【図 1 3】



【図 1 4】

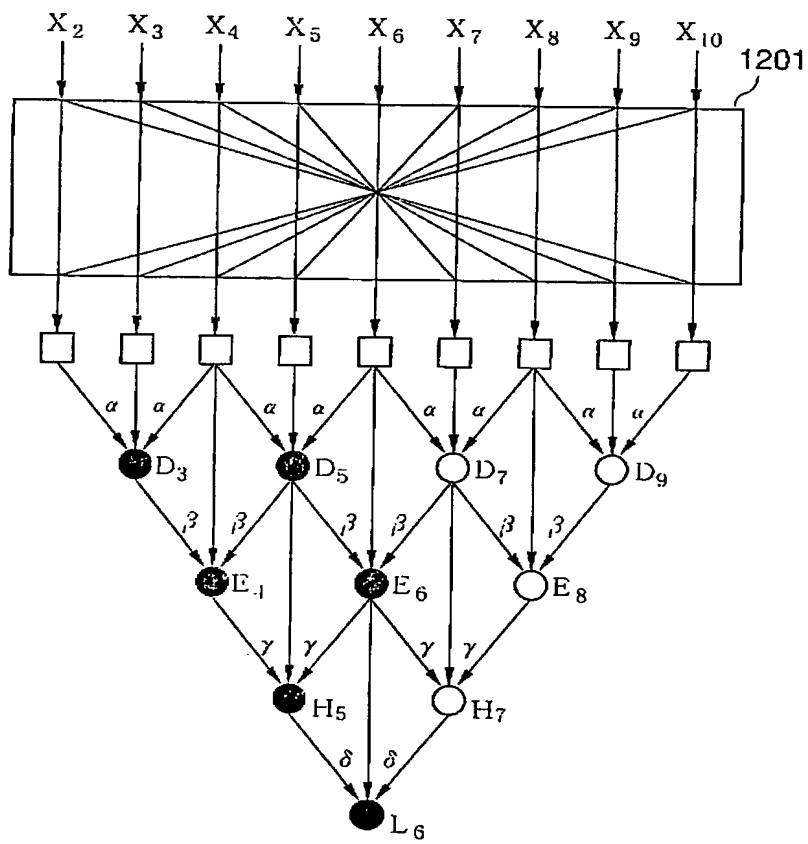


【図 1 5】

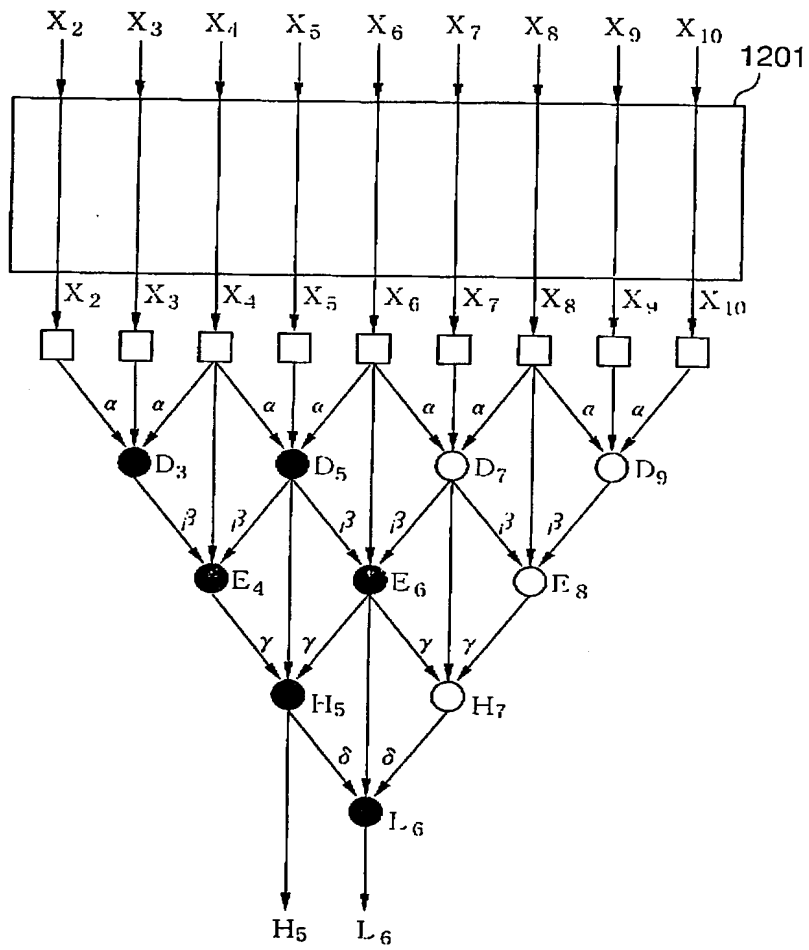




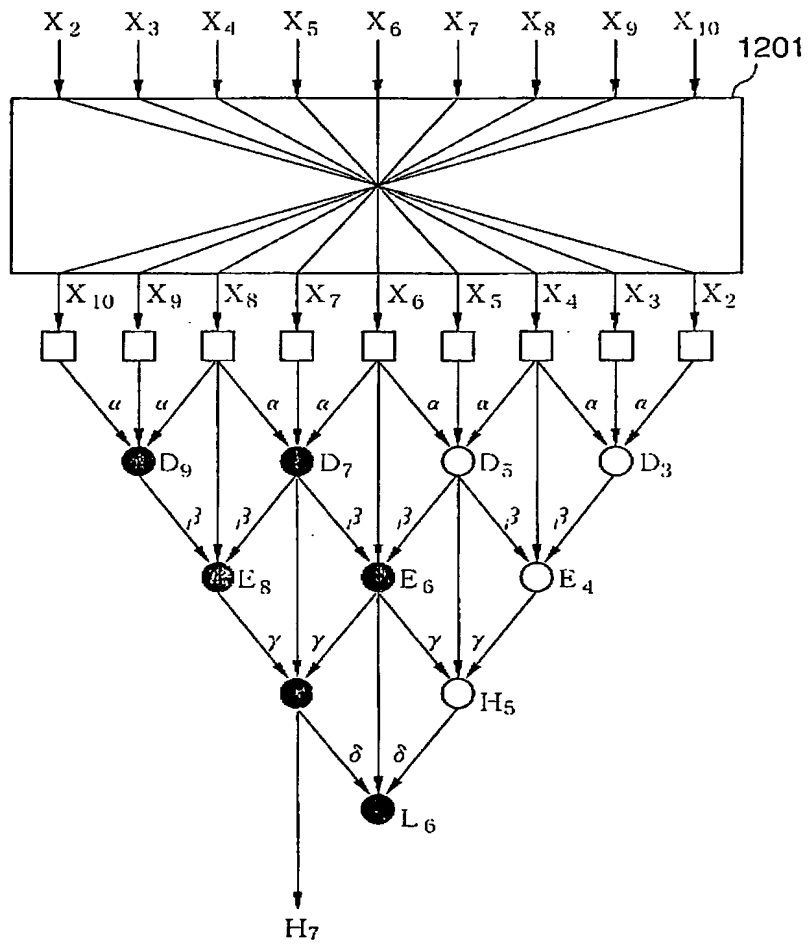
【図 1 6】



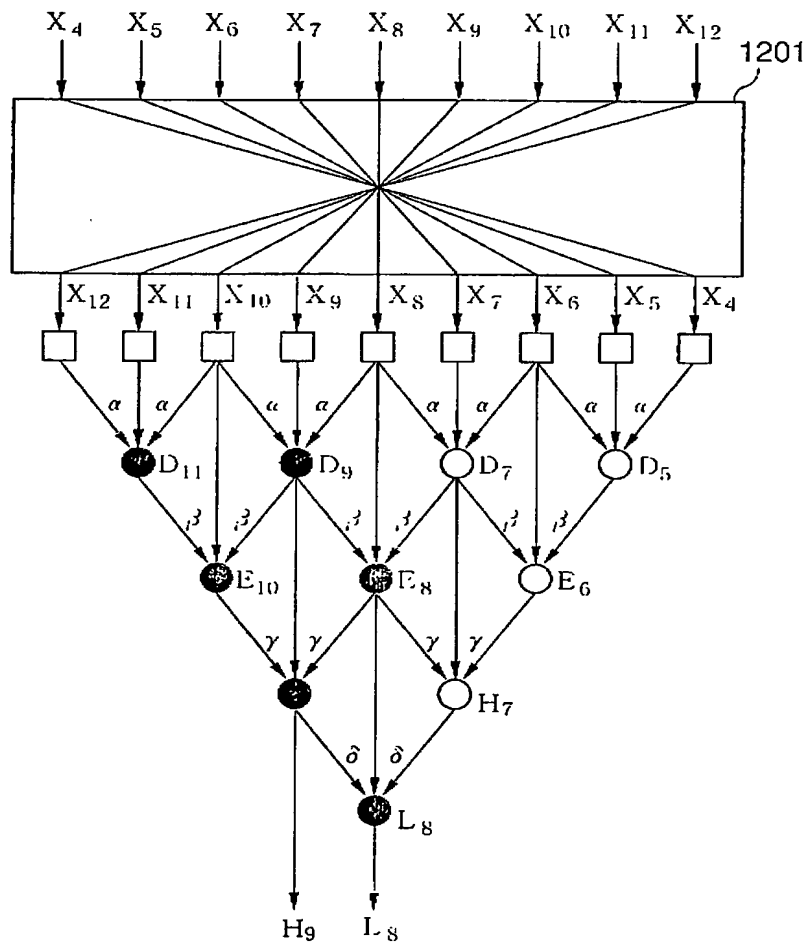
【図 1 7】



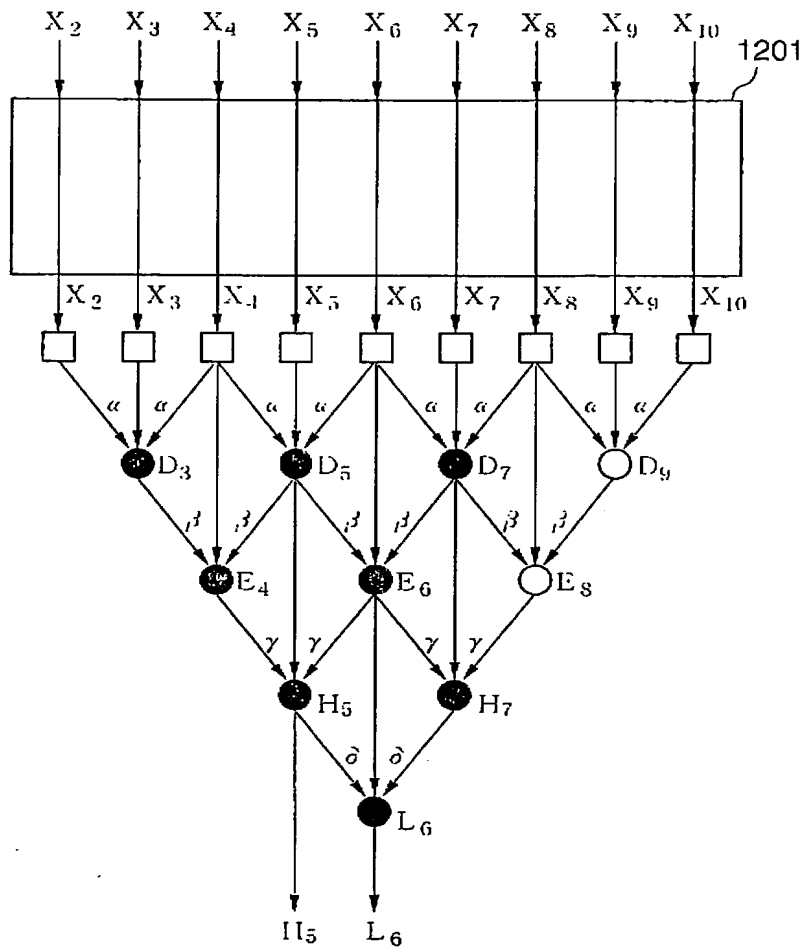
【図 1 8】



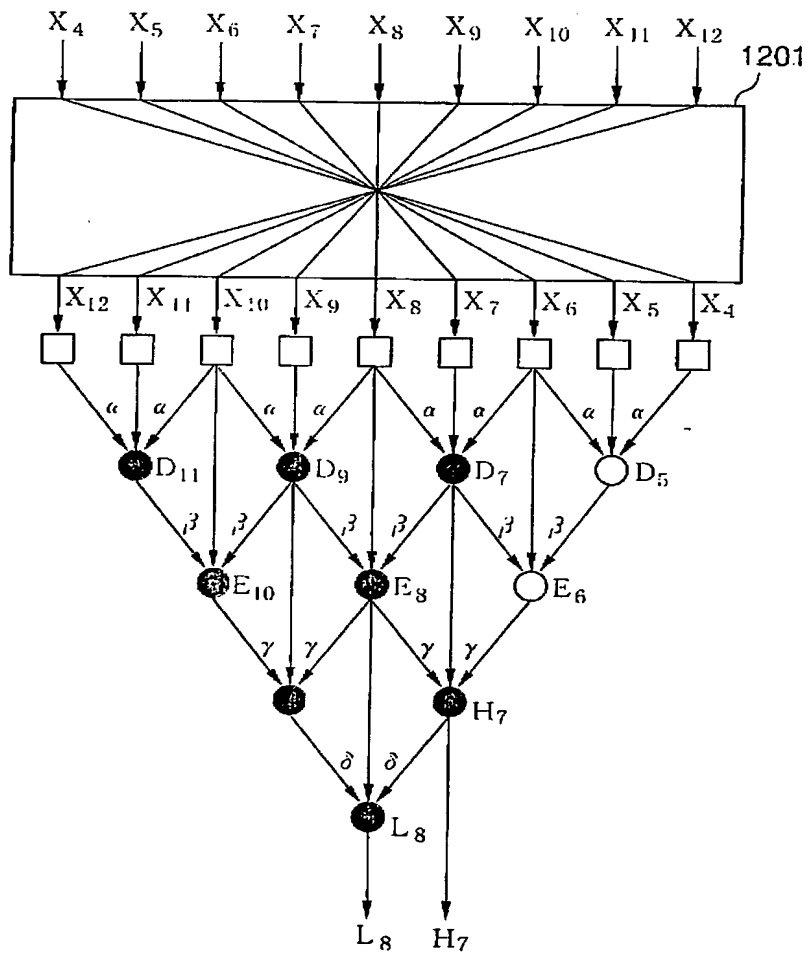
【図 1 9】



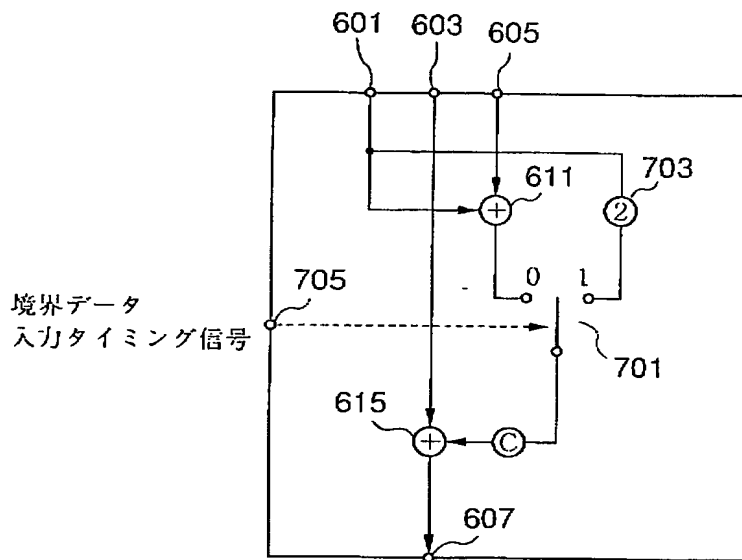
【図 2 0】



【図 2 1】



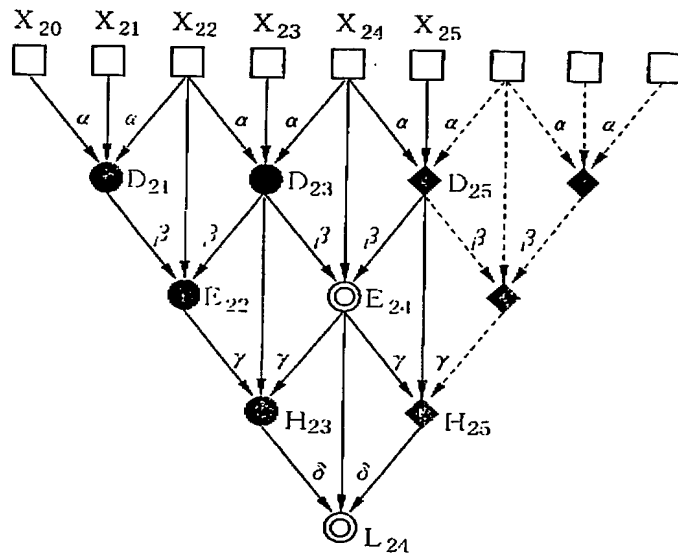
【図 2 2】



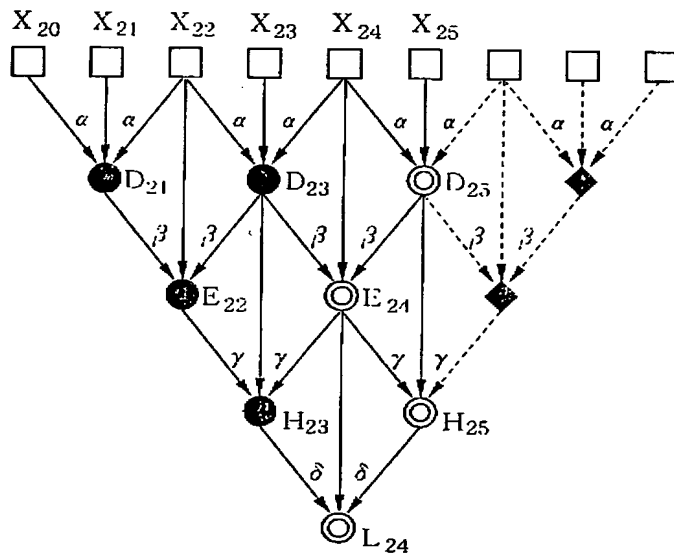




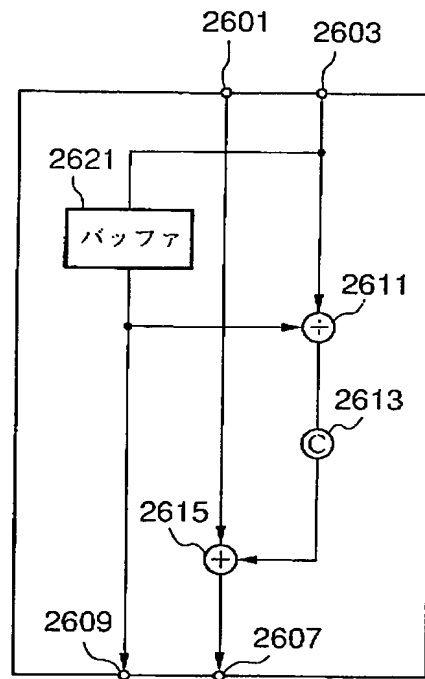
【図 2 4】



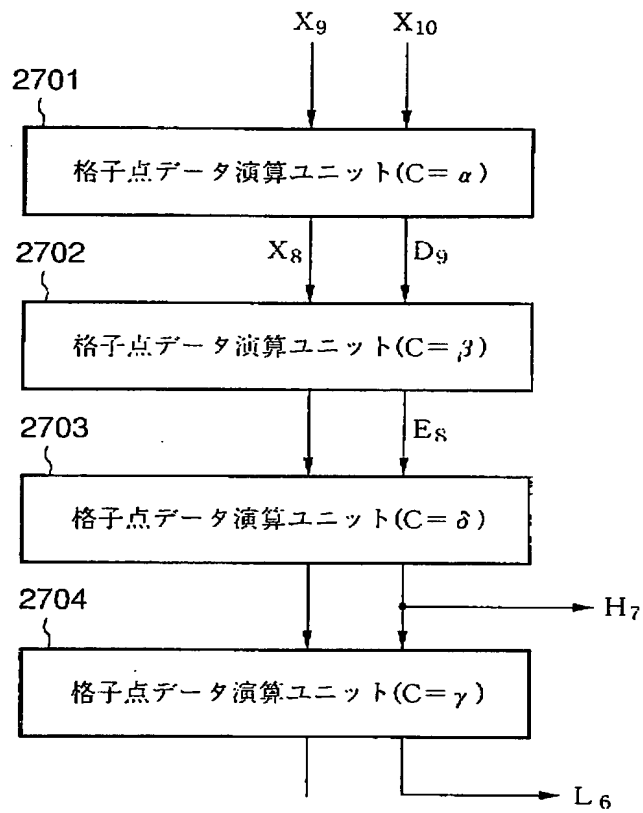
【図 2 5】



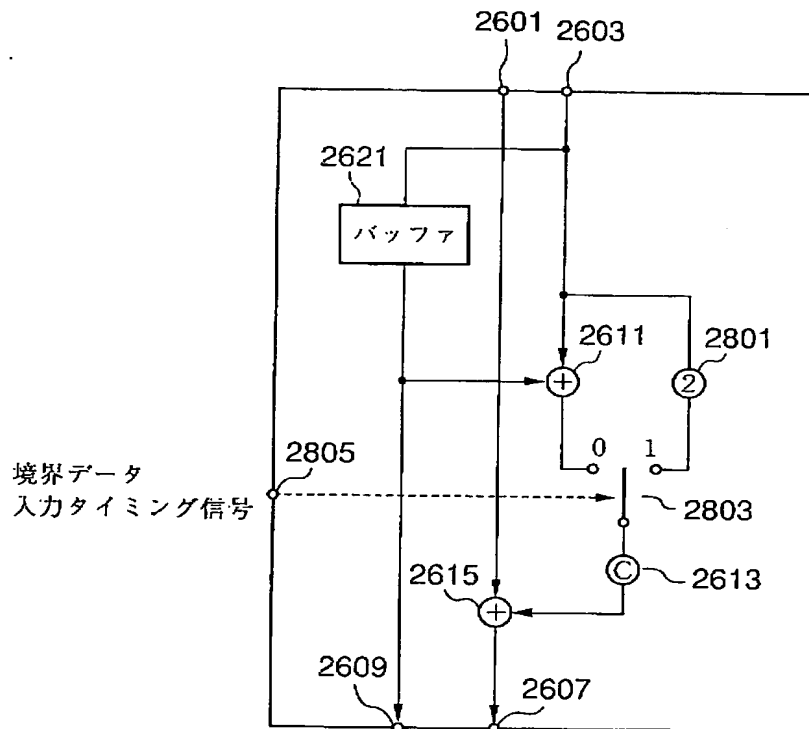
【図 2 6】



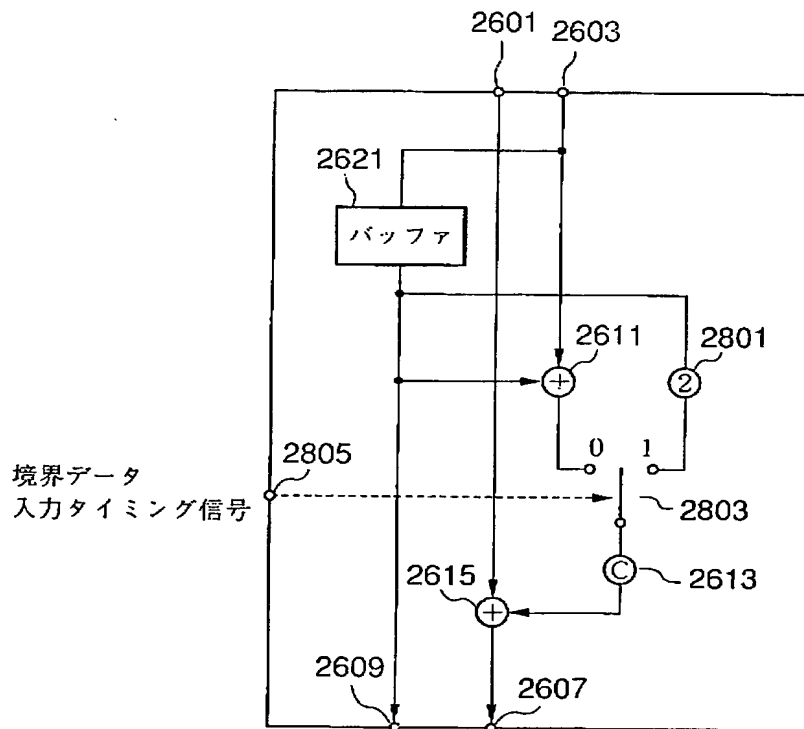
【図 2 7】



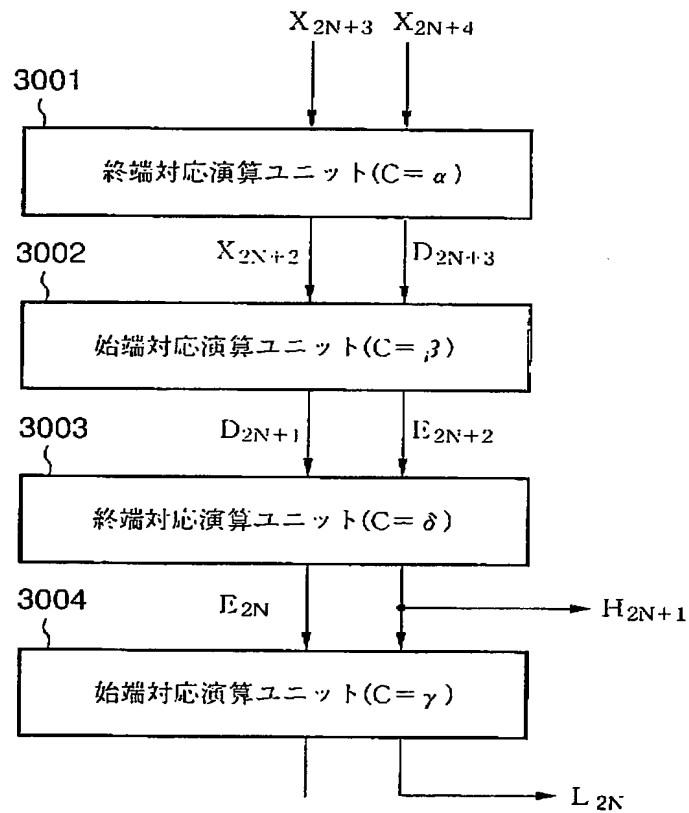
【図 2 8】



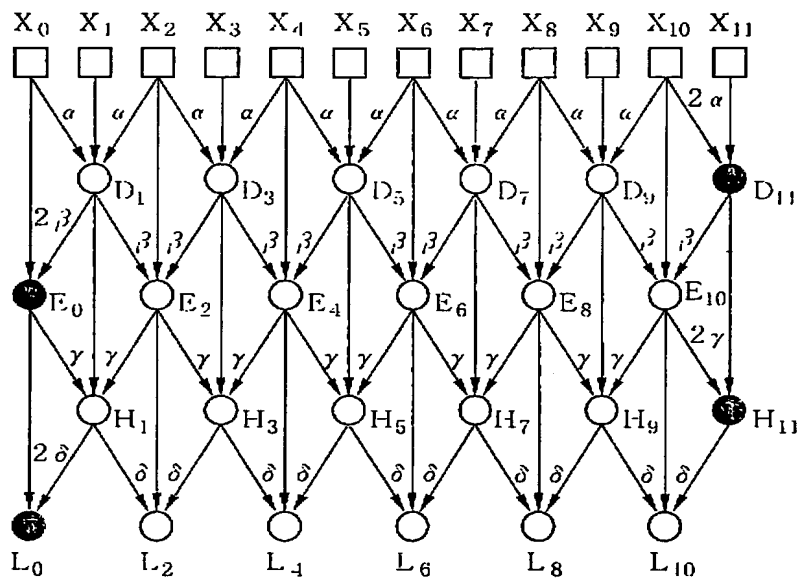
【図 2 9】



【図 3 0】

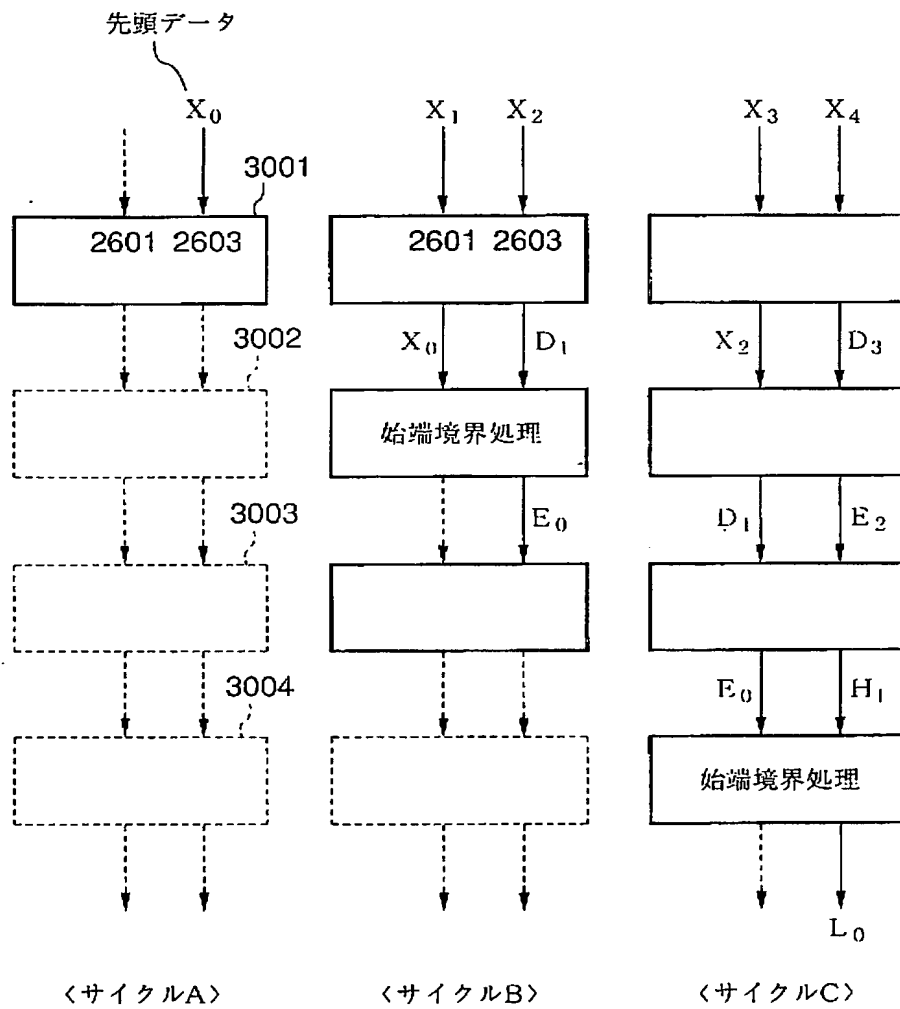


【図 3 1】

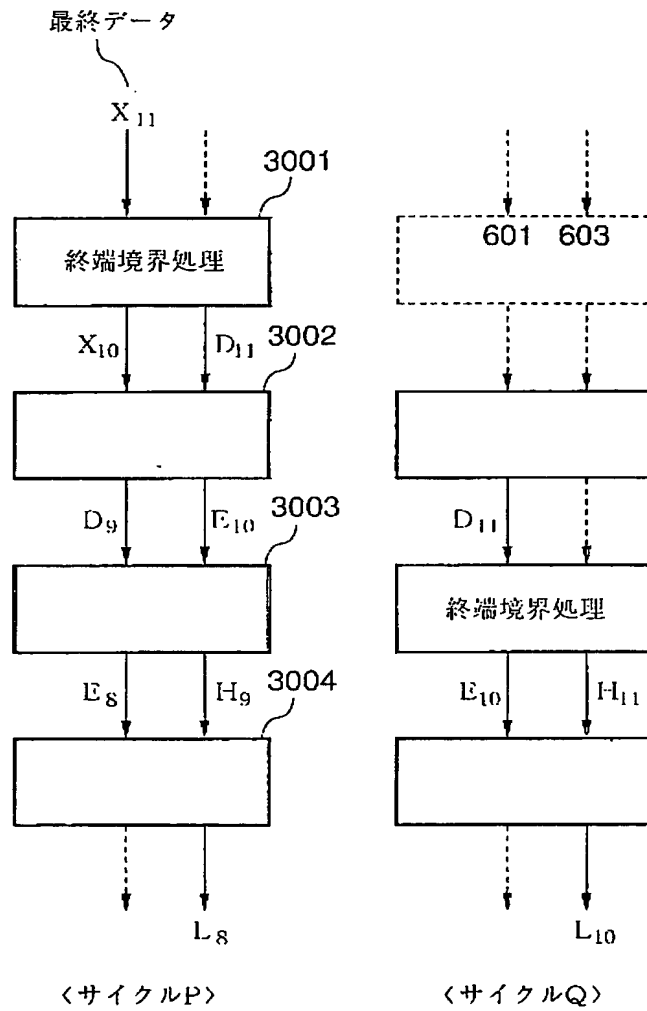




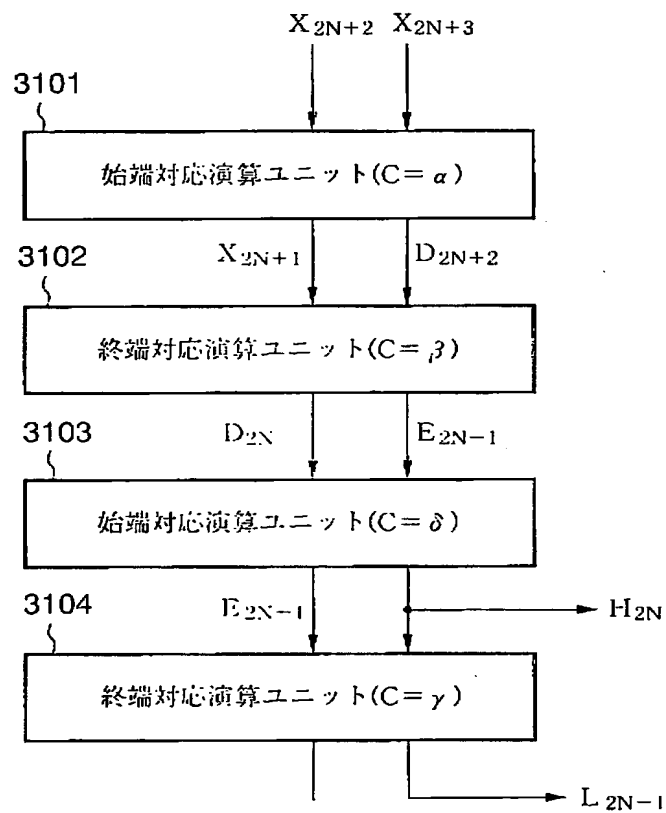
【図 3 2】



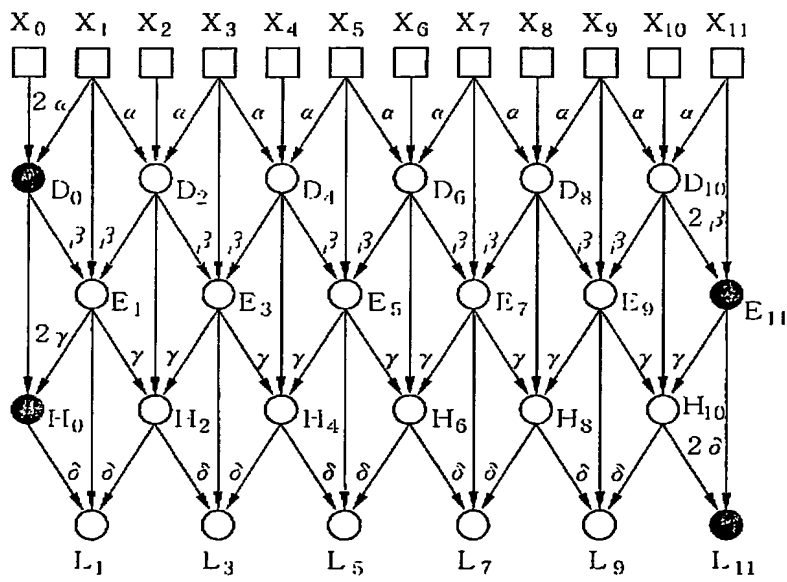
【図 3 3】



【図 3 4】

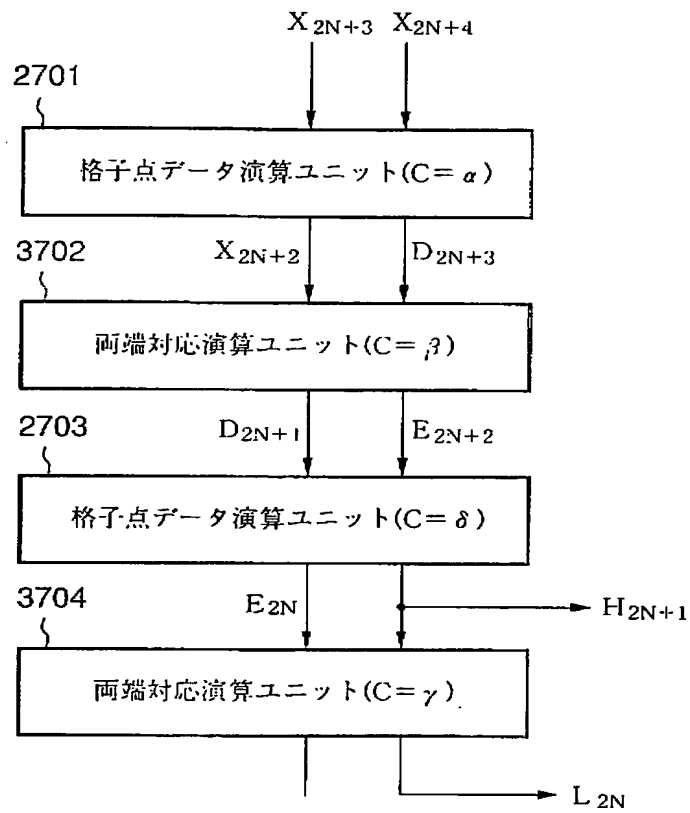


【図 3 5】

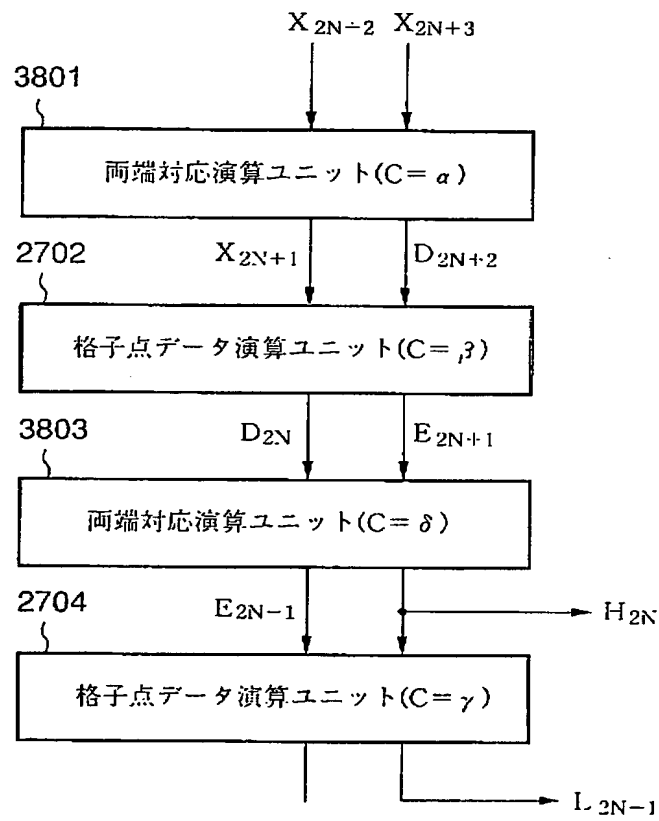




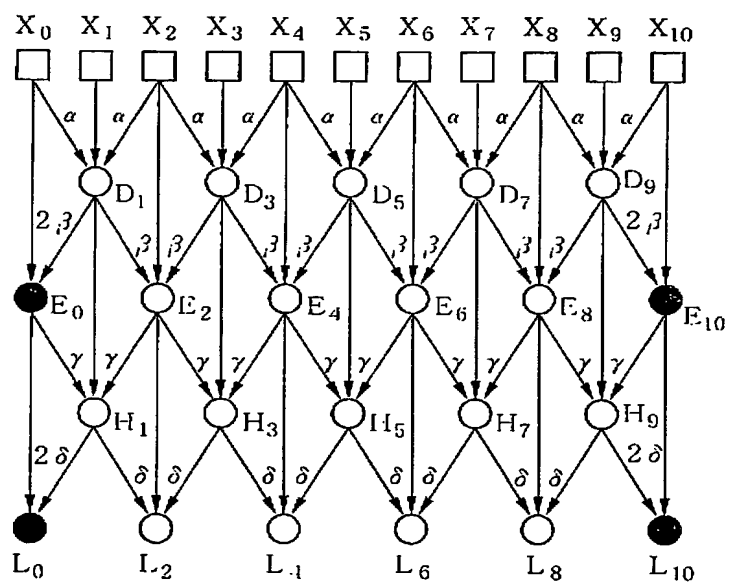
【図 3 7】



【図 3 8】

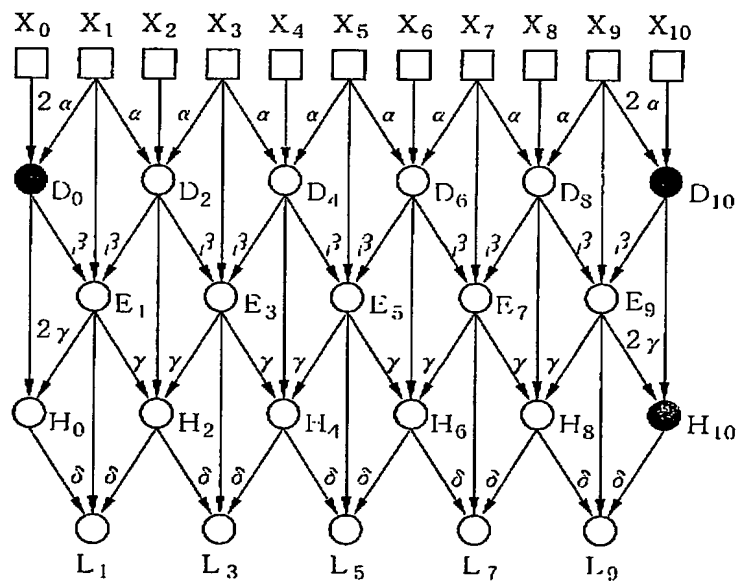


【図 3 9】

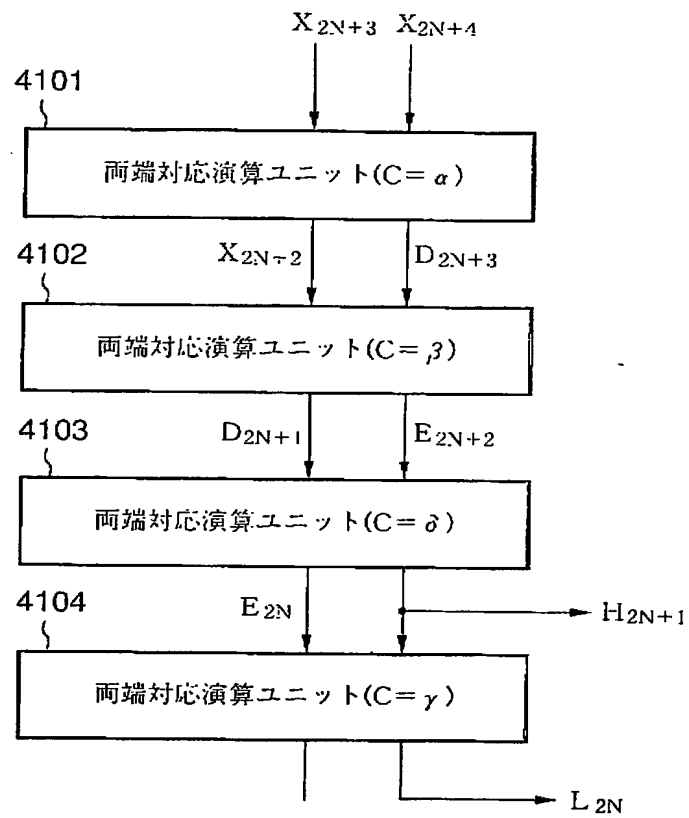




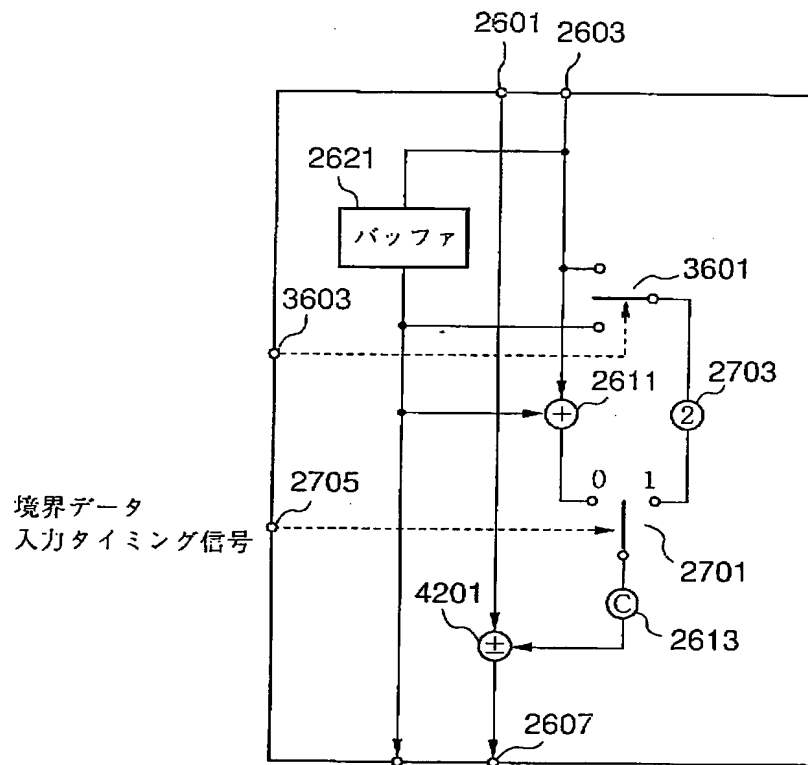
【図 4 0】



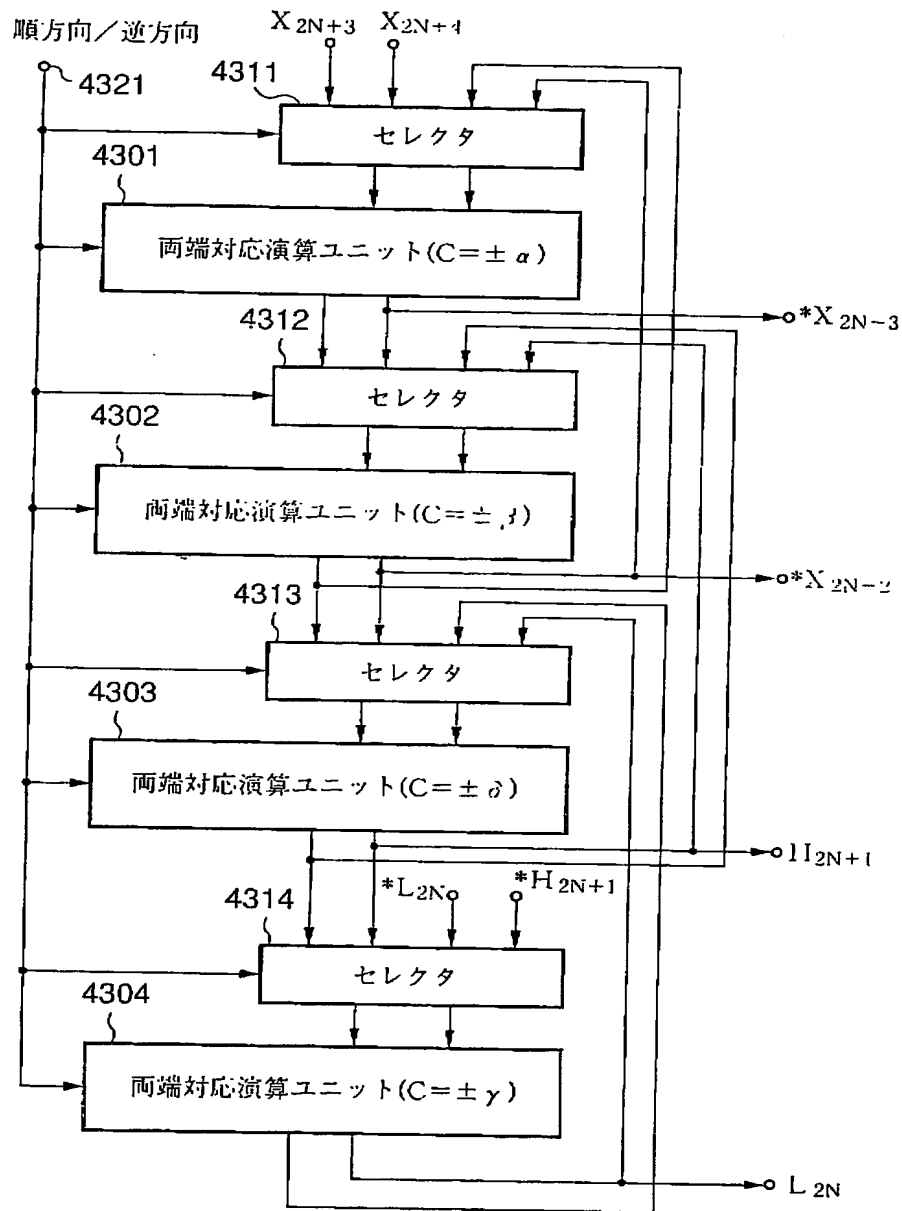
【図 4 1】



【図 4 2】



【図 4 3】



【書類名】 要約書

【要約】

【課題】 処理しようとするデータが画像データの境界外にある場合であっても、その境界外のデータを前処理で生成する必要が無く、以って構造を簡略化させる。

【解決手段】 3つのデータを入力して出力データを算出する演算ユニットを多数・多階層に接続してフィルタ処理装置を構成する際に、その中の幾つかを3入力で1つの出力データを算出するモードと、3入力の中の2つを用いて出力データを算出するモードとが切り換わるように構成する。2入力から出力データを算出するのは、画像の境界の演算タイミングで機能するようにする。

【選択図】 図7

出 願 人 履 歴 情 報

識別番号 [000001007]

1. 変更年月日 1990年 8月30日

[変更理由] 新規登録

住 所 東京都大田区下丸子3丁目30番2号  
氏 名 キヤノン株式会社